

UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR



**Grado en Ingeniería de Tecnologías y Servicios de
Telecomunicación**

TRABAJO FIN DE GRADO

**APLICACIÓN ANDROID SOBRE MAPEO DE FUNCIONES
LÓGICAS EN LOOK-UP TABLES**

Daniel Gómez García
Tutor: Eduardo Boemo Scalvinoni

Junio 2017

APLICACIÓN ANDROID SOBRE MAPEO DE FUNCIONES LÓGICAS EN LOOK-UP TABLES

AUTOR: Daniel Gómez García
TUTOR: Eduardo Boemo Scalvinoni

Laboratorio de Sistemas Digitales
Dpto. Tecnología Electrónica y de Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio de 2017



Resumen (castellano)

Este Proyecto Fin de Grado tiene como objetivo entrenar a estudiantes de ingeniería en la idea conceptual del mapeo de funciones a través de Look-Up Tables. Para ello se propone una colección de ejercicios que pueden resolverse de forma interactiva y comprobarse mediante una aplicación Android.

Los problemas propuestos sobre Look-Up Tables corresponden a las asignaturas “Circuitos Electrónicos Digitales” y “Dispositivos Integrados Especializados” impartidas en el primer y tercer curso del Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación de la Escuela Politécnica Superior en la Universidad Autónoma de Madrid.

Los conceptos que abarca son: a) Operadores lógicos básicos; b) Funciones lógicas combinacionales más comunes (codificadores, decodificadores), c) Funciones aritméticas básicas (suma, resta, multiplicación y división) y finalmente, d) Generación de funciones y operadores matemáticos complejos como seno, coseno, logaritmo, potenciación y raíz de grado n .

La versión final se llama Look-Up Tables (LUTs). Es gratuita y puede descargarse desde Google Play Store.



Abstract (English)

This End-Of-Grade Project aims to train engineering students in the conceptual idea of mapping functions using Look-Up Tables. In order to achieve this, we propose an exercise collection and each exercise can be solved interactively and checked using an Android application.

The posed problems on the Look-Up Tables app are placed on the subjects “Digital Electronic Circuits” and “Specialized Integrated Devices” taught in the first and third year of the Degree in Telecommunication Technology and Service Engineering of the Superior Polytechnic School at the Autónoma University of Madrid.

The concepts covered are: a) Basic logic operators b) Combinational logic functions (Encoder, Decoders) c) Arithmetic functions (addition, subtraction, multiplication and division) and finally, d) Generation of complex mathematical functions and operators such as sine, cosine, logarithm, grade n .

The final version is called Look-Up Tables (LUTs). It's free and can be downloaded from the Google Play Store for any Android device.

Palabras clave (castellano)

Look-Up Table, luts, circuitos, electrónica, digital, móvil, multiplexor, puertas lógicas, aplicación, memoria rom, bcd-7 segmentos, funciones matemáticas, Android.

Keywords (inglés)

Look-Up Table, luts, circuits, electronics, digital, smartphone, multiplexer, logical gates, application, rom memory, bcd-7segments, mathematics functions, Android.

Agradecimientos

Me gustaría agradecer a mi familia por su apoyo incondicional durante todos estos años.

A mi compañera de viaje, el mejor regalo que me ha dado esta carrera.

A mi tutor, Eduardo Boemo, por haberme permitido realizar este TFG mostrándome su confianza y apoyo en todo momento.

A mis amigos, por todos esos momentos y experiencias vividas juntos.

“Soy el amo de mi destino, soy el capitán de mi alma.” Nelson Mandela

“Imposible significa que no has encontrado la solución.” Henry Ford

INDICE DE CONTENIDOS

1 INTRODUCCIÓN.....	1
1.1 MOTIVACIÓN	1
1.2 OBJETIVOS.....	1
1.3 ORGANIZACIÓN DE LA MEMORIA	2
2 ESTADO DEL ARTE	3
2.1 DIGITAL ELECTRONICS 101	3
2.2 LOGICAL GATES	3
2.3 DIGITAL ELECTRONICS AND LOGIC	3
2.4 BOOLEAN LAB	4
2.5 RESUMEN.....	4
3 DISEÑO	5
3.1 FINALIDAD.....	5
3.2 REQUISITOS	5
3.2.1 Elección de sistemas operativos.....	5
3.2.2 Versiones de Android	8
3.2.3 Compatibilidad de dispositivos	10
3.2.4 Compatibilidad de pantallas	10
3.2.5 Idioma de la aplicación	13
3.3 ESTRUCTURA DE LA APLICACIÓN	14
3.3.1 Diagrama de bloques de la aplicación	14
3.3.2 Menú principal	15
3.3.3 Tutorial.....	15
3.3.4 Ejercicios	15
3.3.5 Comenzar, cargar o eliminar ejercicio.....	16
3.3.6 Comprobar	16
3.3.7 Solucionar	16
3.3.8 Resetear	17
3.3.9 Salvar	17
3.3.10 Enviar ejercicios.....	17
3.3.11 Ayuda	18
3.3.12 Menú Android	18
3.3.13 Ejercicios de la aplicación.....	18
4 DESARROLLO.....	25
4.1 PRIMEROS PASOS	25
4.2 HERRAMIENTAS	25
4.3 PARTES DE UNA APLICACIÓN.....	25
4.3.1 Actividad	25
4.3.2 Vistas.....	27
4.3.3 La carpeta Manifests.....	27
4.3.4 La carpeta Java	28
4.3.5 La carpeta Recursos	28
4.3.6 La carpeta Gradle.scripts	29
4.4 MENÚS DE LA APLICACIÓN	29
4.4.1 Tutorial.....	30
4.4.2 Acerca de.....	30
4.5 DISEÑO Y DESARROLLO DE LOS EJERCICIOS	31
4.5.1 Vista de la actividad	31
4.5.2 Actividad	32
4.6 GESTIÓN DE DATOS	33

4.6.1 Método de preferencias	33
4.6.2 Check	33
4.6.3 Solve	33
4.6.4 Reset	34
4.6.5 Save	34
4.6.6 Load or delete	35
4.6.7 Send exercises	36
4.7 ADAPTACIÓN DE PANTALLAS.....	36
5 INTEGRACIÓN, PRUEBAS Y RESULTADOS	37
5.1 GOOGLE PLAY STORE.....	37
5.2 PREPARACIÓN	37
5.3 CUENTA DE DESARROLLADOR.....	39
5.4 PUBLICACIÓN	39
6 CONCLUSIONES Y TRABAJO FUTURO.....	41
6.1 CONCLUSIONES.....	41
6.2 TRABAJO FUTURO	42
REFERENCIAS	43
GLOSARIO	45
ANEXOS	- 1 -
A CLASES Y MÉTODOS JAVA.....	- 1 -
B IMÁGENES DE LOS EJERCICIOS DE LA APLICACIÓN	- 3 -
C FUENTES BIBLIOGRÁFICAS.....	- 5 -

INDICE DE FIGURAS

FIGURA ESTADO DEL ARTE - 1: TABLA COMPARATIVA.....	4
FIGURA REQUISITOS - 1: TABLA COMPARATIVA ENTRE SO DE SMARTPHONES.....	6
FIGURA REQUISITOS - 2: PROGRESIÓN DESDE 1ST CUATRIMESTRE DE 2009 HASTA EL 1ST DE 2017.	7
FIGURA REQUISITOS - 3: DISTRIBUCIÓN VERSIONES ANDROID	8
FIGURA REQUISITOS - 4: PROGRESIÓN VERSIONES ANDROID	9
FIGURA REQUISITOS - 5: EJEMPLO DE MEDIDAS EN PANTALLAS	10
FIGURA REQUISITOS - 6: TAMAÑOS DE PANTALLAS USADOS EN DISPOSITIVOS ANDROID	11
FIGURAS REQUISITOS - 7: DENSIDADES DE PANTALLAS USADAS EN DISPOSITIVOS ANDROID	11
FIGURA REQUISITOS - 8: DENSIDADES Y TAMAÑOS DE PANTALLAS AGOSTO 2016	12
FIGURA REQUISITOS - 9: DENSIDADES Y TAMAÑOS DE PANTALLAS NOVIEMBRE 2016	12
FIGURA REQUISITOS - 10: TOP TEN LENGUAJES EN INTERNET.	13
FIGURA ESTRUCTURA DE LA APLICACIÓN - 11: DIAGRAMA DE BLOQUES DE LA APLICACIÓN	14
FIGURA DESARROLLO – PARTES DE UNA ACTIVIDAD - 1: CICLO DE VIDA DE UNA ACTIVIDAD ANDROID	26
FIGURA DESARROLLO – CARPETA MANIFESTS - 2: EJEMPLO DE ANDROIDMANIFEST.XML	27
FIGURA DESARROLLO – CARPETA JAVA- 3: EJEMPLO DE MAINACTIVITY.JAVA	28
FIGURA DESARROLLO – MENÚS DE LA APLICACIÓN - 4: MENÚS DE LA APLICACIÓN.....	29
FIGURA DESARROLLO – TUTORIAL - 5: TUTORIAL DE LA APLICACIÓN	30
FIGURA DESARROLLO – ACERCA DE - 6: ABOUT DE LA APLICACIÓN	30
FIGURA VISTA DE LA ACTIVIDAD - 7: APARIENCIA DE LOS EJERCICIOS DE LA APLICACIÓN	32
FIGURA GESTIÓN DE DATOS - CHECK - 8: MENSAJE DE COMPROBACIÓN DE EJERCICIO.....	33
FIGURA GESTIÓN DE DATOS - SOLVE - 9: MENSAJE DE MOSTRAR SOLUCIÓN EN EJERCICIO.....	33
FIGURA GESTIÓN DE DATOS - RESET - 10: MENSAJE DE RESETEO DE EJERCICIO	34
FIGURA GESTIÓN DE DATOS - SAVE - 11: MENSAJE DE GUARDADO DE EJERCICIO	34
FIGURA GESTIÓN DE DATOS – LOAD OR DELETE - 12 : MENSAJE DE CARGAR UN EJERCICIO	35

FIGURA GESTIÓN DE DATOS – LOAD OR DELETE - 13: MENSAJE DE CARGAR O ELIMINAR UN EJERCICIO.....	35
FIGURA GESTIÓN DE DATOS – SEND EXERCISES - 14: PROCESO DE ENVÍO DE UN EJERCICIO.	36
FIGURA INTEGRACIÓN, PRUEBAS Y RESULTADOS– PREPARACIÓN - 1: PROCESO DE GENERACIÓN APK	37
FIGURA INTEGRACIÓN, PRUEBAS Y RESULTADOS– PREPARACIÓN - 2: PROCESO DE INSTALACIÓN ARCHIVO APK	39

INDICE DE TABLAS

TABLA EJERCICIOS DE LA APLICACIÓN 1: EJERCICIO 1	18
TABLA EJERCICIOS DE LA APLICACIÓN 2: EJERCICIO 2	19
TABLA EJERCICIOS DE LA APLICACIÓN 3: EJERCICIO 3	19
TABLA EJERCICIOS DE LA APLICACIÓN 4: EJERCICIO 4.....	20
TABLA EJERCICIOS DE LA APLICACIÓN 5: EJERCICIO 5	20
TABLA EJERCICIOS DE LA APLICACIÓN 6: EJERCICIO 6.....	21
TABLA EJERCICIOS DE LA APLICACIÓN 7: EJERCICIO 7	21
TABLA EJERCICIOS DE LA APLICACIÓN 8: EJERCICIO 8.....	22
TABLA EJERCICIOS DE LA APLICACIÓN 9: EJERCICIO 9	22
TABLA EJERCICIOS DE LA APLICACIÓN 10: EJERCICIO 10.....	23
TABLA EJERCICIOS DE LA APLICACIÓN 11: EJERCICIO 11	23
TABLA EJERCICIOS DE LA APLICACIÓN 12: EJERCICIOS 12, 13 Y 14	24

1 Introducción

1.1 Motivación

Este proyecto surge siguiendo la línea de otras aplicaciones del DSLab, para acercar a los alumnos las ventajas que los smartphones hoy en día nos ofrecen, ayudándoles con su aprendizaje sobre problemas de las asignaturas de “Circuitos Electrónicos Digitales” y “Dispositivos Integrados Especializados”.

Algunas de las ventajas que podríamos destacar son:

- Accesibilidad al material en cualquier momento y lugar.
- No es necesaria la conexión a Internet.
- Sustitución del formato impreso de las guías de problemas.
- Corrección instantánea de los problemas.
- Comparación y/o envío de los resultados con otros compañeros o el profesor.
- Comodidad y visualización.

1.2 Objetivos

El objetivo de este Trabajo de Fin de Grado es desarrollar una aplicación Android que sirva como herramienta de aprendizaje a los alumnos de las asignaturas “Circuitos Electrónicos Digitales” y “Dispositivos Integrados Especializados” de la Escuela Politécnica Superior de la Universidad Autónoma de Madrid.

La idea principal del proyecto es facilitar su uso para el análisis y comprensión de las Look-Up Tables con diferentes problemas de distinta complejidad.

Los alumnos podrán realizar las opciones de guardar y cargar los ejercicios, comprobación de los mismos, y la opción de envío para compartirlos con otros alumnos o con el profesor en caso de tener dudas al respecto sobre algún ejercicio específico.

El resultado de este proyecto es un complemento de ayuda además de las guías de problemas tradicionales, pudiendo analizar y resolver estos ejercicios que pertenecen al temario de la asignatura de manera interactiva.

Los conceptos generales de los ejercicios sobre Look-Up Tables de esta aplicación son los siguientes:

- Operadores lógicos básicos.
- Funciones lógicas combinacionales más comunes (codificadores, decodificadores).
- Funciones aritméticas básicas (suma, resta, multiplicación y división).
- Generación de funciones y operadores matemáticos complejos como seno, coseno, logaritmo, potenciación y raíz de grado n .

1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

- **Capítulo 1: Introducción.**
- **Capítulo 2: Estado del arte.**
- **Capítulo 3: Diseño.**
- **Capítulo 4: Desarrollo.**
- **Capítulo 5: Integración, pruebas y resultado.**
- **Capítulo 6: Conclusiones y trabajo futuro.**

2 Estado del arte

El objetivo de este proyecto es la creación de una aplicación móvil a través de la cual los alumnos puedan mejorar sus conocimientos relacionados con las Look-Up Tables.

Las aplicaciones disponibles en Google Play Store no cubren el tema propuesto. A continuación, se incluye un estudio de trabajos similares (Electrónica Digital) ya sean de pago o gratuitas que aborden conceptos desarrollados en esta aplicación.

2.1 Digital Electronics 101

Esta aplicación está disponible de manera gratuita en Google Play Store, aunque también tiene versión de pago. Ofrece una colección muy completa con un temario sobre conceptos básicos, puertas lógicas, circuitos lógicos, multiplexores, en definitiva, un temario de utilidad sobre Electrónica Digital.

Incluye descripciones bastante detalladas, y la dinámica de la aplicación de los temas que ofrece es la de test con opción múltiple.

Modelo teórico para aprender conceptos básicos. No hay opción de completar los ejercicios de manera manual.

2.2 Logical Gates

Esta aplicación está disponible de manera gratuita en Google Play Store. Ofrece un modelo teórico y limitado ya que sólo nos muestra las distintas puertas lógicas con sus respectivas tablas de verdad sin posibilidad de proponerse como problema.

No dispone de tutorial ni la opción de envío de ejercicios.

2.3 Digital Electronics and Logic

Esta aplicación está disponible de manera gratuita en Google Play Store. Ofrece un modelo teórico, pero con un mayor temario disponible, ya que además de las puertas lógicas con sus respectivas tablas de verdad, entre sus contenidos también están sumadores, codificadores, multiplexores.

Todo se muestra mediante una breve descripción con un esquema o tabla de verdad correspondiente. No hay opción a ejercicios.

2.4 Boolean Lab

Esta aplicación está disponible de manera gratuita en Google Play Store. Ofrece un modelo teórico, incluyendo en todos los casos propuestos la opción de seleccionar las dos distintas entradas en las puertas lógicas solucionando las correspondientes salidas.

Además, ofrece la posibilidad de dibujar la onda de la señal, algo interesante para los alumnos al ser de ayuda para comprender conceptos de la asignatura. También incluye flip-flops, y una variedad de ejercicios con la dinámica de variar las dos entradas a los multiplexores.

No dispone de envío de ejercicios ni de tutorial.

Esta aplicación podría ser una introducción de conceptos a la aplicación propiamente desarrollada.

2.5 Resumen

Después de haber realizado el estudio de aplicaciones similares en la web, podemos llegar a la conclusión de que no existe ninguna con el mismo temario que proponga ejercicios con mayor complejidad y variedad que la desarrollada.

Ya que la aplicación más parecida en ciertos aspectos sólo dispone de multiplexores de dos entradas, y ninguna aplicación tiene la opción disponible del envío de ejercicios.

Otra de las aplicaciones más adecuada por contenido sólo dispone de test.

En la siguiente tabla se muestra una breve comparativa con las aplicaciones analizadas:

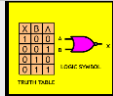


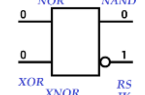

APP	Tutorial	Test	Ejercicios	Envío Soluciones	Icono
<i>Digital Electronics 101</i>	Si	Si	No	No	
<i>Logical Gates</i>	No	No	No	No	
<i>Digital Electronics and Logic</i>	Si	No	No	No	
<i>Boolean Lab</i>	No	No	Si	No	
<i>Look-Up Tables</i>	Si	No	Si	Si	

Figura Estado del arte - 1: Tabla comparativa

3 Diseño

3.1 Finalidad

Este proyecto tiene por objetivo facilitar a los estudiantes de manera interactiva la resolución de problemas relacionados con las Look-Up Tables.

La aplicación consta de un tutorial, una lista formada por 14 ejercicios diferentes, y diversas opciones como son la comprobación de las soluciones, consejos sobre dónde pueden estar los fallos de las respuestas, y la opción del envío de los ejercicios al profesor o distintos compañeros en caso de tener dudas con algo en concreto.

Se pretende ayudar a los alumnos de una forma más cercana a las nuevas tecnologías y de manera personal para que puedan aprender cómodamente y en cualquier parte sólo con su dispositivo móvil o Tablet.

3.2 Requisitos

3.2.1 Elección de sistemas operativos

En primer lugar, comprobamos las distintas opciones que el mercado hoy en día nos ofrece y nos centraremos en los más utilizados por los dispositivos:

- ***Windows 10 Mobile***

Tercer sistema operativo más usado en el mercado, anteriormente llamado Windows Phone. El cambio iba destinado a llegar a más dispositivos móviles con la llegada de Windows 10 a los ordenadores. Pertenece a Microsoft y es de código cerrado.

- ***IOS***

Se sitúa en segundo lugar como sistema operativo más usado a nivel mundial. Pertenece a Apple Inc y es de código cerrado. Utilizado en los dispositivos de la gama Apple.

- ***Android***

Cuenta con el mayor número de instalaciones de smartphones de todo el Mundo. Primera plataforma móvil completa de software libre y código abierto. En España tiene un 90% de mercado a fecha de Marzo 2017 según KantarWorldPanel.

Se ha elegido como sistema operativo para el desarrollo de la aplicación, Android, ya que cualquier herramienta que necesitamos de desarrollo es libre. En nuestro caso, hemos usado para nuestro proyecto Android Studio, software gratuito y multiplataforma.

Además, las aplicaciones se pueden usar en dispositivos con distintos sistemas operativos ya sean Linux, Mac OS X y Windows. Sin ser necesaria una cuenta de desarrollador, o con un coste de US25\$ en el caso de querer publicarla en Google Play Store.

Mientras que en IOS sólo pueden ser usados en dispositivos con Mac OS X y la publicación en App Store tiene un coste de US99\$ al año, y la cuenta de desarrollador es libre con Xcode 7.

Al realizar el estudio de mercado comprobamos la progresión de los sistemas operativos mencionados anteriormente:

Period	Android	iOS	Windows Phone	Others
2015Q4	79.6%	18.7%	1.2%	0.5%
2016Q1	83.5%	15.4%	0.8%	0.4%
2016Q2	87.6%	11.7%	0.4%	0.3%
2016Q3	86.8%	12.5%	0.3%	0.4%

Figura Requisitos - 1: Tabla comparativa entre SO de smartphones.¹

Observamos como Android es el sistema operativo para smartphones más usado en el mercado, mientras el resto de sistemas operativos pierden relevancia.

En la siguiente tabla veremos una progresión más detallada de la evolución de los distintos sistemas operativos según los millones de unidades vendidos:

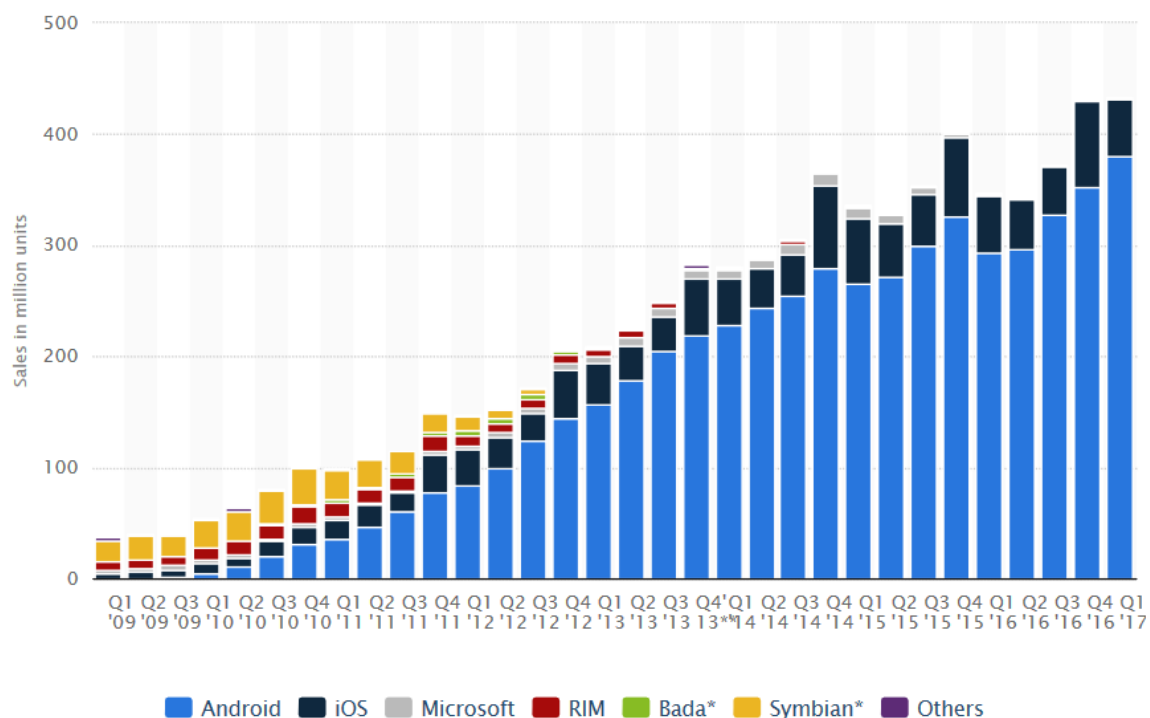


Figura Requisitos - 2: Progresión desde 1st cuatrimestre de 2009 hasta el 1st de 2017. ²

Tras observar la tendencia evolutiva, Android aumenta sus ventas respecto a sus competidores, llegando a un mayor número de dispositivos. A la vez que se comprueba que los dos sistemas operativos con importancia son IOS y Android, perdiendo la misma otros sistemas operativos con el paso del tiempo.

Motivo por el cual es de gran importancia, ya que, aunque en un principio la aplicación está pensada para los alumnos de la Escuela Politécnica Superior, al desarrollar la aplicación en Android nos aseguramos las opciones de poder llegar a un mayor número de personas.

Como hecho relevante en cuanto a sistemas operativos se refiere, en la actualidad, Fuchsia, anteriormente llamado Andrómeda, es un nuevo sistema operativo con el que Google pretende sustituir a Android.

Cabe destacar que ya no dependerá del kernel de Linux pudiendo mejorar su sistema operativo a su manera.

3.2.2 Versiones de Android

Cada versión del sistema operativo tiene su propio nivel de API. Cuanta más baja sea la versión de API, más dispositivos podrán instalar la aplicación, sin embargo, si elegimos una versión demasiado baja no podremos utilizar muchos elementos de Android introducidos por las versiones posteriores.

La web oficial de Android nos proporciona datos sobre la cantidad de dispositivos distribuidos y sus versiones instaladas, pero al no partir de datos anteriores no podemos hacer una estimación:

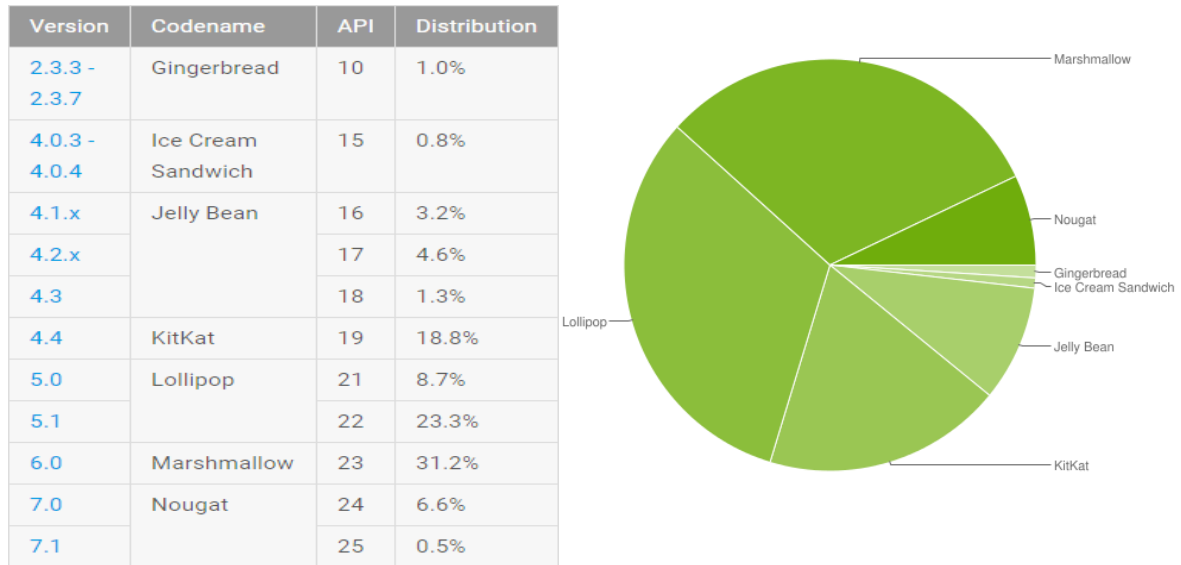


Figura Requisitos - 3: Distribución versiones Android ³

Viendo estas imágenes pensaríamos que hay un claro dominio en la distribución de las APIs Marshmallow y Lollipop en los dispositivos, pero tenemos que mirar un período de tiempo actualizado para intentar conocer la tendencia de esas distribuciones.

Como hemos mencionado, vamos a observar la tendencia de las APIS a lo largo de un período de tiempo para poder sacar conclusiones:

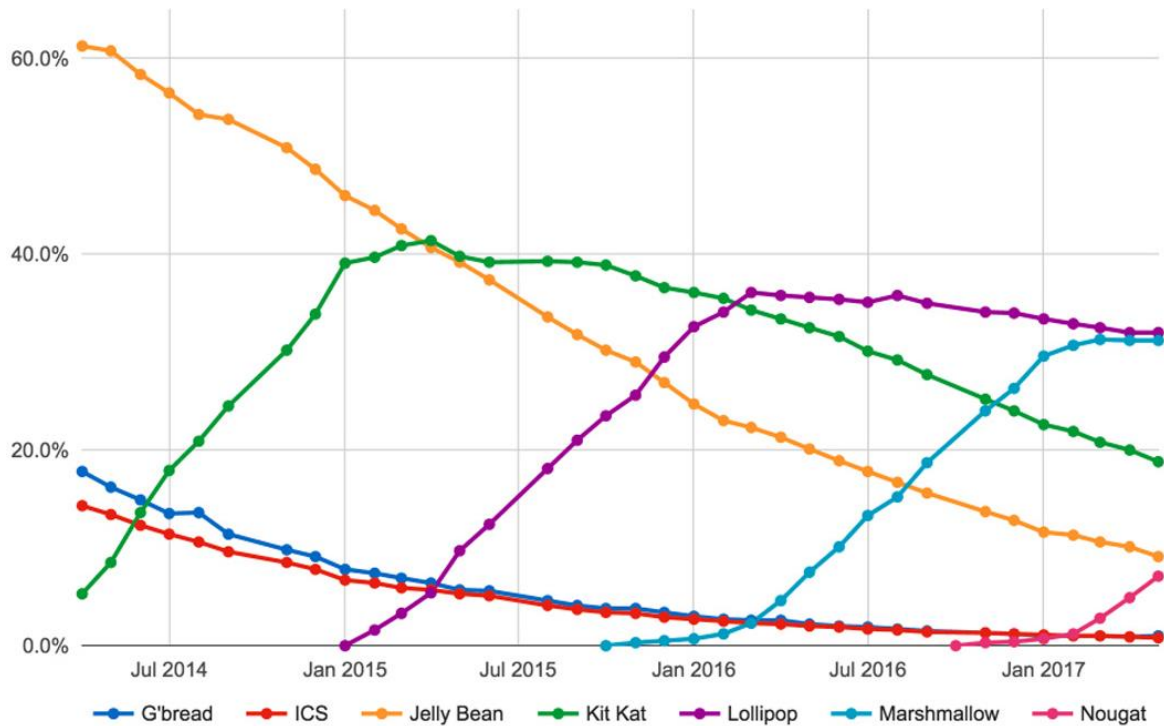


Figura Requisitos - 4: Progresión versiones Android ⁴

Podemos observar que Android Nougat, desde su aparición lleva una clara tendencia alcista sobre todo con la nueva gama de dispositivos que llegan al mercado con esta versión instalada de fábrica.

Kit Kat y Jelly Bean por el contrario llevan una clara tendencia bajista, aunque se mantienen tercera y cuarta versión respectivamente más usada, mientras que Marshmallow se destaca como segunda versión más usada del sistema operativo por detrás sólo de Lollipop.

Tras los resultados observados se optó por diseñar la aplicación para dispositivos con una versión mínima de Ice Cream Sandwich y versión máxima de Nougat.

3.2.3 Compatibilidad de dispositivos

Hoy en día la competitividad de los smartphones es muy superior a las de las tablets ya que hay smartphones con pantallas más grandes y ordenadores más ligeros.

El uso de smartwatch tampoco está indicado para el desarrollo de esta aplicación al ser en principio realizada para la docencia.

3.2.4 Compatibilidad de pantallas

El sistema operativo Android se ejecuta en múltiples dispositivos con diferentes tamaños de pantalla y distintas densidades.

Tenemos que tratar de optimizar los cambios, los ajustes de tamaños y las diferentes densidades de cara a que el usuario lo interprete como que la aplicación está diseñada para su móvil en exclusividad y pueda transmitir un resultado satisfactorio de la misma.

Las principales categorías en la compatibilidad de pantallas son:

- Tamaño de pantalla.

Tamaño físico real de la pantalla, cuya longitud se considera la diagonal de la misma.

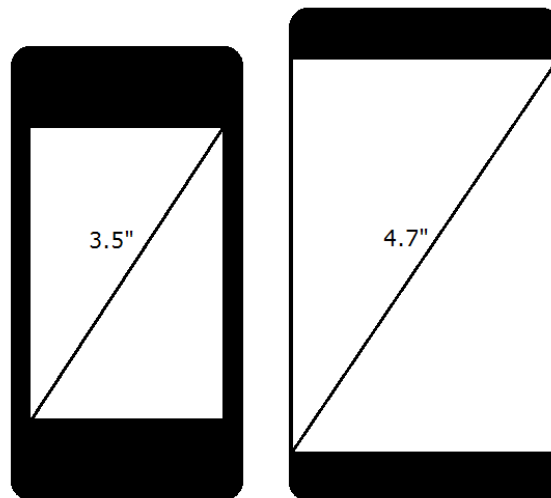


Figura Requisitos - 5: Ejemplo de medidas en pantallas

Android reconoce 4 tipos de pantalla medidos por el número de pulgadas de la diagonal:

- Pequeño (small) de 2 a 3 pulgadas.
- Normal (normal) de 3 a 4.5 pulgadas.
- Grande (large) de 4.5 a 7 pulgadas.
- Extragrande(xlarge)de 7 a 10 pulgadas.

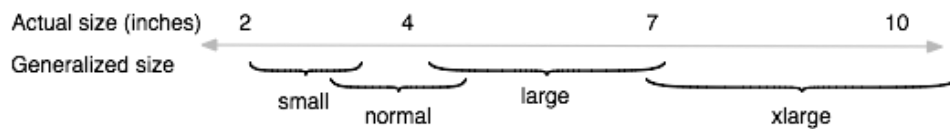
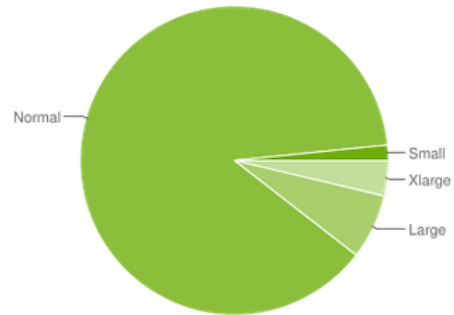
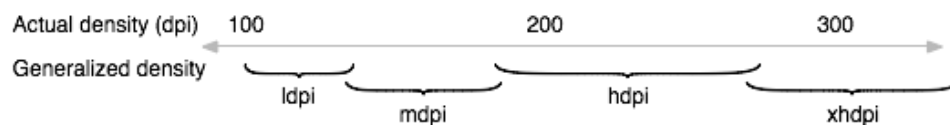
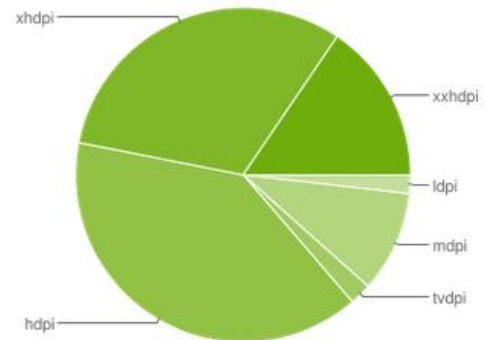


Figura Requisitos - 6: Tamaños de pantallas usados en dispositivos Android ⁵

- Densidad de pantalla.

Número de píxeles en un área física de la pantalla. Se mide en puntos por pulgada (dpi). Y según el número de píxeles por área se clasifican en:

- Baja (ldpi) ~ 120 dpi.
- Media (mdpi) ~ 160 dpi.
- Alta (hdpi) ~ 240 dpi.
- Extra alta (xhdpi) ~ 320 dpi.
- Extra Extra alta (xxhdpi) ~ 480 dpi.
- Extra Extra Extra alta (xxxhdpi) ~ 640 dpi.



Figuras Requisitos - 7: Densidades de pantallas usadas en dispositivos Android ⁶

- Orientación.

Orientación de la pantalla desde la visión del usuario, ya sea orientación horizontal u orientación vertical. Lo que se considera relación de aspecto a lo ancho o a lo alto.

- Resolución.

Número total de píxeles en la pantalla.

- Píxeles independientes de la densidad.

Número de píxeles virtuales que hay que definir en el diseño de la interfaz de usuario.

En las siguientes tablas encontraremos a modo de resumen el porcentaje de dispositivos con los diferentes tamaños y densidades más utilizadas en un período de tiempo para poder compararlos:

	ldpi	mdpi	tvdpi	hdpi	xhdpi	xxhdpi	Total
Small	1.8%						1.8%
Normal		3.8%	0.1%	40.0%	27.3%	15.5%	86.7%
Large	0.2%	4.3%	2.1%	0.5%	0.5%		7.6%
Xlarge		2.9%		0.3%	0.7%		3.9%
Total	2.0%	11.0%	2.2%	40.8%	28.5%	15.5%	

Figura Requisitos - 8: Densidades y tamaños de pantallas Agosto 2016 ⁷

	ldpi	mdpi	tvdpi	hdpi	xhdpi	xxhdpi	Total
Small	1.6%						1.6%
Normal		3.1%	0.2%	38.7%	30.4%	15.5%	87.9%
Large	0.2%	3.9%	1.9%	0.4%	0.4%		6.8%
Xlarge		2.8%		0.3%	0.6%		3.7%
Total	1.8%	9.8%	2.1%	39.4%	31.4%	15.5%	

Figura Requisitos - 9: Densidades y tamaños de pantallas Noviembre 2016 ⁸

Tras los estudios realizados, empezando por los tamaños de las pantallas más usadas en los dispositivos, el más usado es el tamaño normal, con mucha diferencia respecto de los otros tipos.

De hecho, es el único de los distintos tipos de tamaño que ha aumentado su porcentaje respecto los demás, que han disminuido.

Luego hemos diseñado la aplicación para el tamaño normal. También se ha decidido incluir el tamaño grande (large), ya que, aunque se haya reducido el porcentaje, es el segundo con mayor uso en los dispositivos Android.

En cuanto a las densidades en los dispositivos, la baja (ldpi), media (mdpi) y la densidad alta (hdpi) descienden relativamente sus porcentajes.

La extra alta (xxhdpi) si tiene un aumento considerable, casi un 3% respecto unos meses atrás. Esto es beneficioso de cara al usuario ya que mejora la calidad de las imágenes de una manera más nítida y detallada.

Hemos desarrollado la aplicación con una densidad alta (hdpi), porque, aunque haya disminuido algo su porcentaje sigue siendo la más usada en los dispositivos. También hemos incluido la extra alta (xxhdpi) y la extra extra alta (xxxhdpi) ya que son las siguientes densidades más usadas con porcentajes importantes.

3.2.5 Idioma de la aplicación

El idioma elegido para desarrollar la aplicación ha sido el inglés, porque, aunque es superado por otros idiomas como el chino, o el español en cuanto a hablantes de lengua materna se refiere, es el idioma más utilizado en el mundo en Internet.

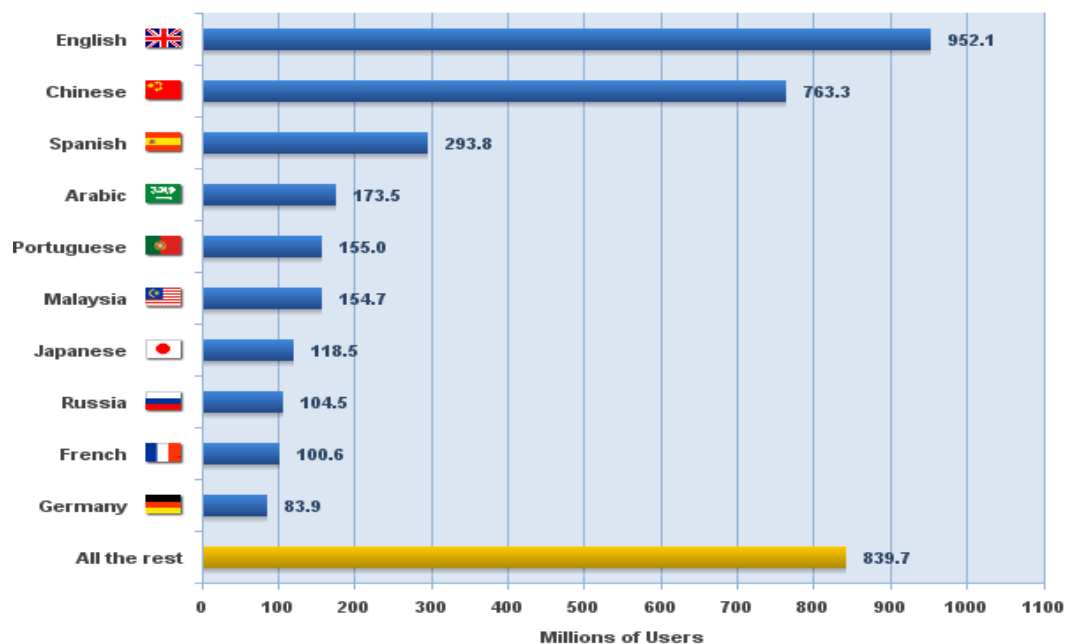


Figura Requisitos - 10: Top ten lenguajes en Internet. ⁹

3.3 Estructura de la aplicación

3.3.1 Diagrama de bloques de la aplicación

En cuanto a la estructura de la aplicación, se ha decidido mantener la estructura del DSLab para dar solidez y concordancia con el resto de aplicaciones del laboratorio.

Al estar enfocada a los alumnos de la escuela, lo que se pretende es que sea lo más fácil de usar y comprender, por ello, se ha estructurado del siguiente modo:

Un menú principal desde el que se puede acceder a todas las funciones de la aplicación que son las siguientes, un tutorial sobre las Look-Up Tables, la lista con los distintos ejercicios propuestos para los alumnos, la opción de envío de los mismos en caso de tener dudas en concreto con algún ejercicio, un botón de ayuda para explicaciones más concisas, y otro botón detallando las características de la versión y programación de la misma.

De un modo gráfico quedaría de la siguiente manera:

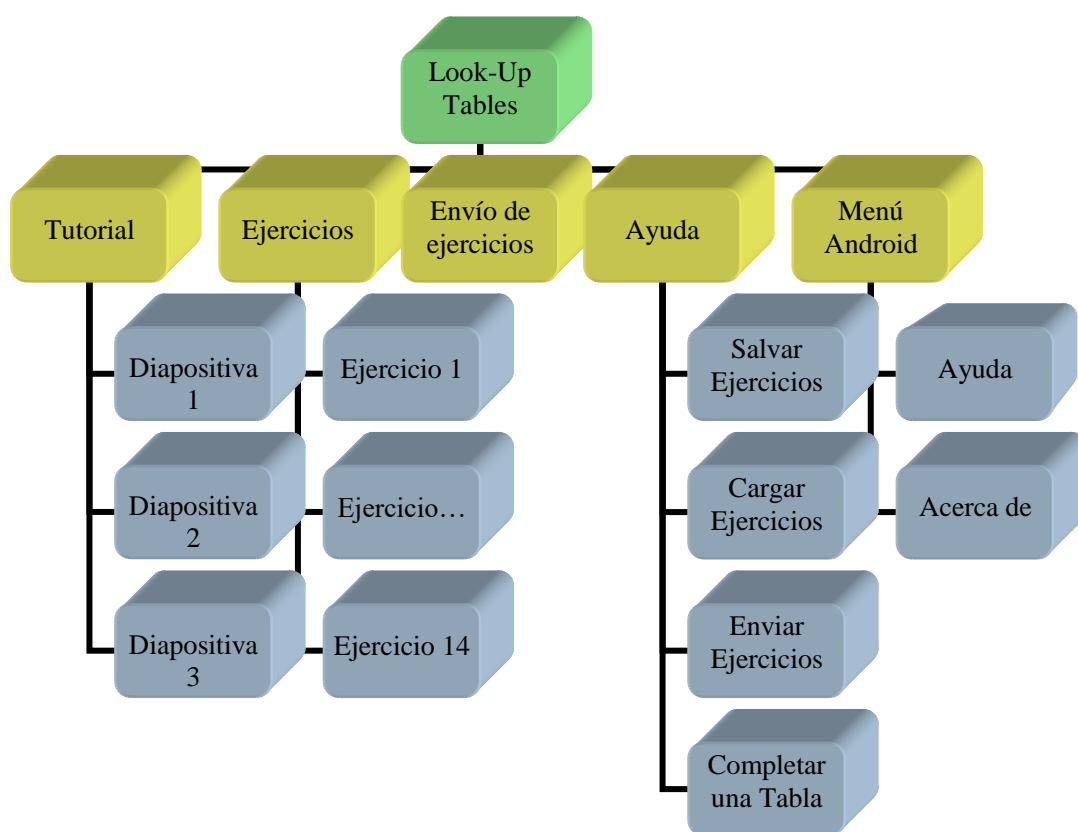


Figura Estructura de la aplicación - 11: Diagrama de bloques de la aplicación

3.3.2 Menú principal

Desde el menú principal de la aplicación podremos acceder a las siguientes opciones:

- Tutorial.
- Ejercicios.
- Envío de Ejercicios.
- Ayuda.
- Menú Android.

3.3.3 Tutorial

En este apartado, se incluirá un tutorial para facilitar al estudiante unas explicaciones generales acerca de las Look-Up Tables de cara a que no le surjan dudas a la hora de realizar los ejercicios. Constará de varias diapositivas en las que para avanzar o retroceder bastará con deslizar el dedo en la pantalla hacia la izquierda si queremos avanzar o hacia la derecha para retroceder de diapositiva.

Para salir de este apartado será suficiente con pulsar el botón de atrás del móvil Android o back.

El tutorial con sus distintas imágenes será diseñado por el profesor de la asignatura Circuitos Electrónicos Digitales.

3.3.4 Ejercicios

Al acceder a la sección de ejercicios, nos lleva a otro menú donde se encuentra la lista de los diferentes ejercicios.

El total de ejercicios que forman la lista son 14, orientados de menor a mayor complejidad o dificultad de cara a facilitar el aprendizaje del estudiante.

Se han intentado dividir en lo que serían 3 grandes bloques, el primero de ellos sería para que los alumnos asentarán los conocimientos vistos al comienzo del curso de la asignatura sobre las distintas puertas lógicas, y sepan completar las conexiones en una LUT de 2 entradas actuando como un multiplexor. Es un comienzo para que refuercen los conceptos sobre las puertas lógicas con sus correspondientes tablas de verdad y cómo realizar las conexiones.

El segundo bloque se centraría en mejorar los conocimientos de los ejercicios propuestos anteriormente mediante la actuación de una LUT de 3 entradas con diferentes problemas.

Y el último bloque tendría como objetivo el que los alumnos puedan abordar problemas de una LUT de 4 entradas actuando como una memoria ROM. Es muy importante que comprendan el funcionamiento de la misma para infinidad de recursos posibles.

3.3.5 Comenzar, cargar o eliminar ejercicio

Dentro de cada ejercicio tenemos los respectivos enunciados a los problemas propuestos con sus respectivas imágenes o esquemas, y hay tres opciones disponibles dentro de cada ejercicio:

- Start:

Para dar comienzo al ejercicio.

- Load or Delete:

Para cargar o eliminar algún ejercicio guardado con anterioridad.

En el caso de cargar, sólo podemos cargar un ejercicio, y en caso no de seleccionar ninguno o seleccionar más de uno aparecerá un mensaje de error. Si la carga se realiza con éxito se sobrescribirá a los datos que hubiera antes.

En el caso de eliminar, se pueden eliminar más de un ejercicio a la vez, y nos aparecerá un mensaje para confirmar que queremos eliminar los datos ya que una vez sean eliminados no se podrá, recuperar.

Si pulsamos el botón de start para comenzar la actividad, nos lleva a otra pantalla en la que tenemos visible el esquema del problema propuesto, con los botones de conexión correspondientes para que el alumno muestre sus habilidades en la resolución del mismo.

3.3.6 Comprobar

Dentro de la realización de cada ejercicio, una de las opciones que aparece es la opción “Check”, cuya función es la de comprobar las soluciones aplicadas por el usuario y darle las indicaciones pertinentes mediante la muestra de mensajes para su resolución en caso de que haya algún error en la solución propuesta.

Lo que se pretende con esta opción es la de guiar al estudiante para comprender que está haciendo bien y que no.

3.3.7 Solucionar

Otra de las opciones disponibles dentro de la resolución de un ejercicio es la opción “Solve” que al pulsarla lo que hace es mostrar la solución correspondiente del problema.

Lo que se pretende con esta opción es que el estudiante siempre tenga disponible la solución de todos los problemas para que pueda analizarla en caso de no llegar a la misma. Sobre todo, en los de mayor complejidad.

3.3.8 Resetear

La tercera opción disponible dentro de cada resolución de ejercicio, es la opción “Reset”, cuyo funcionamiento es poner el ejercicio en el estado inicial del mismo.

Antes de ser seleccionada esta opción, aparece un mensaje para asegurar al usuario si quiere o no resetear el ejercicio.

Aunque las conexiones pueden ser cambiadas de estado continuamente sin importar el número de veces que sean pulsadas, lo que se pretende es devolver el problema al estado inicial sin tener ninguna conexión pulsada.

3.3.9 Salvar

Y, por último, la opción disponible en cada resolución de ejercicio es la opción “Save”. Es una opción cuya utilidad es la de salvar los progresos conseguidos de cada alumno.

Los resultados pueden ser guardados para poder ser continuados cargándolos en otro momento. Cuando accedemos a la pulsación aparece un mensaje emergente con la solicitud de un nombre para ser guardar el ejercicio, y en caso de estar repetido da la opción de sobrescribir el mismo.

3.3.10 Enviar ejercicios

Volviendo al menú principal, tenemos disponible la opción “Send exercises”.

La cual, al pulsarla no aparecerá un mensaje emergente para seleccionar los archivos a ser enviados. También nos da la opción de cancelar el envío.

Una vez seleccionados los archivos para enviar, nos despliega un menú con diferentes opciones para el envío del mismo, ya sea a través de correo, bluetooth, mensajería instantánea.

Una vez seleccionado el método de envío, aparecerá el archivo con unas explicaciones sobre dónde será descargado y dónde hay que colocarlo para que al cargarlo funcione correctamente.

3.3.11 Ayuda

En el menú principal también tenemos el botón de ayuda o “Help”. Nos manda a otro menú en el que tenemos diferentes opciones:

- Saving Exercises.
- Loading Exercises.
- Send Exercises.
- Completing a Table.

Según seleccionemos una u otra, tendremos unas explicaciones sobre la opción seleccionada mediante imágenes y notas.

3.3.12 Menú Android

Esta opción se encuentra disponible en todo momento en la aplicación y nos facilita un envío directo a “Help” mencionado anteriormente, o acerca de “About” cuya función es la de mostrar información de la aplicación.

3.3.13 Ejercicios de la aplicación

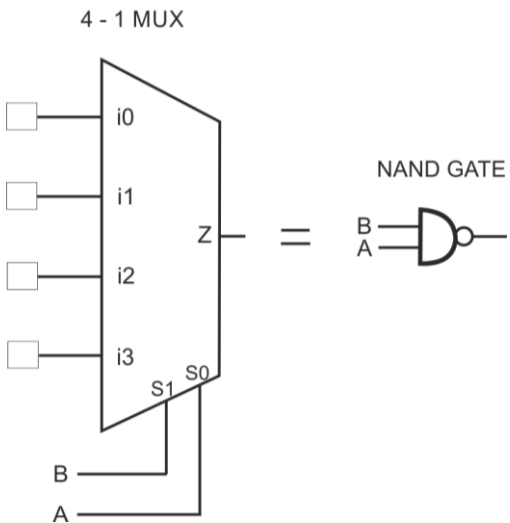
Número: 1
Concepto: Multiplexor 4-1 (2-LUT) como puerta lógica de 2 entradas.
Descripción: Este ejercicio trata sobre la utilización de un multiplexor para realizar la función lógica NAND.
Objetivo: Presentar el tema LUT con los casos más sencillos posibles.
Figura: 

Tabla Ejercicios de la aplicación 1: Ejercicio 1

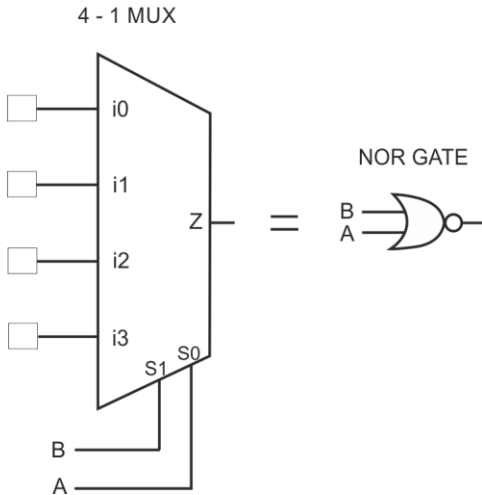
Número: 2
Concepto: Multiplexor 4-1 (2-LUT) como puerta lógica de 2 entradas.
Descripción: Este ejercicio trata sobre la utilización de un multiplexor para realizar la función lógica NOR.
Objetivo: Presentar el tema LUT con los casos más sencillos posibles.
Figura: 

Tabla Ejercicios de la aplicación 2: Ejercicio 2

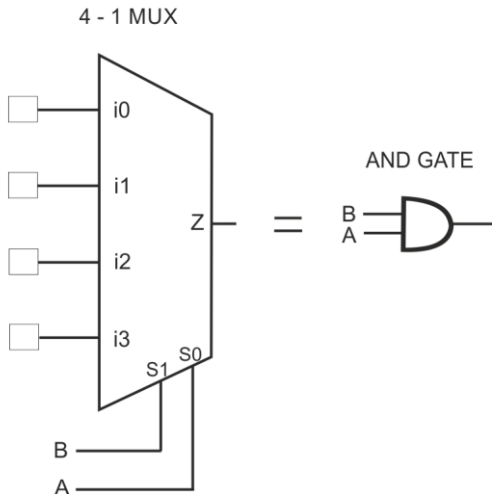
Número: 3
Concepto: Multiplexor 4-1 (2-LUT) como puerta lógica de 2 entradas.
Descripción: Este ejercicio consta de la utilización de un multiplexor para realizar la función lógica AND.
Objetivo: Presentar el tema LUT con los casos más sencillos posibles.
Figura: 

Tabla Ejercicios de la aplicación 3: Ejercicio 3

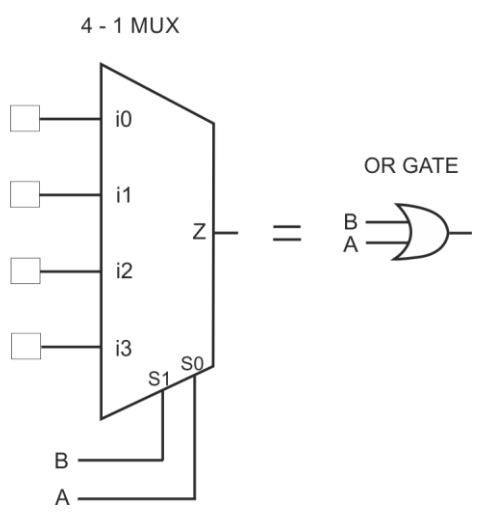
Número: 4
Concepto: Multiplexor 4-1 (2-LUT) como puerta lógica de 2 entradas.
Descripción: Este ejercicio consta de la utilización de un multiplexor para realizar la función lógica OR.
Objetivo: Presentar el tema LUT con los casos más sencillos posibles.
Figura: 

Tabla Ejercicios de la aplicación 4: Ejercicio 4

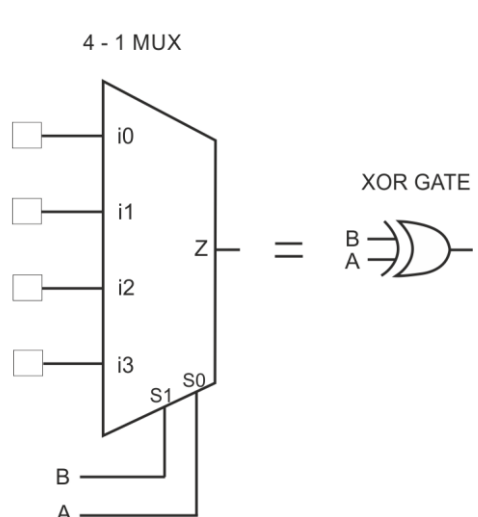
Número: 5
Concepto: Multiplexor 4-1 (2-LUT) como puerta lógica de 2 entradas.
Descripción: Este ejercicio consta de la utilización de un multiplexor para realizar la función lógica XOR.
Objetivo: Presentar el tema LUT con los casos más sencillos posibles.
Figura: 

Tabla Ejercicios de la aplicación 5: Ejercicio 5

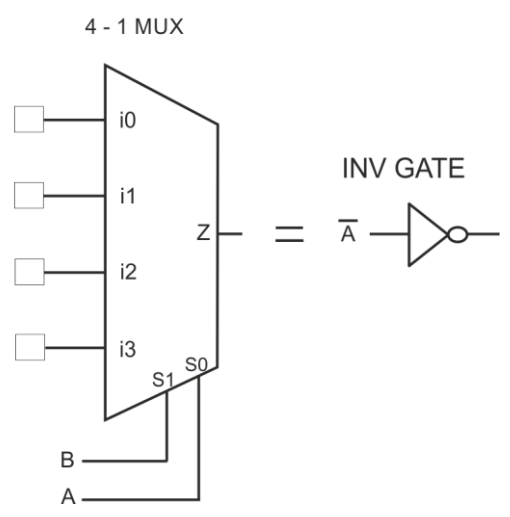
Número: 6
Concepto: Multiplexor para realizar un inversor.
Descripción: Este ejercicio consta de la utilización de un multiplexor para realizar la función lógica INV.
Objetivo: Presentar el tema LUT con los casos más sencillos posibles.
Figura: 

Tabla Ejercicios de la aplicación 6: Ejercicio 6

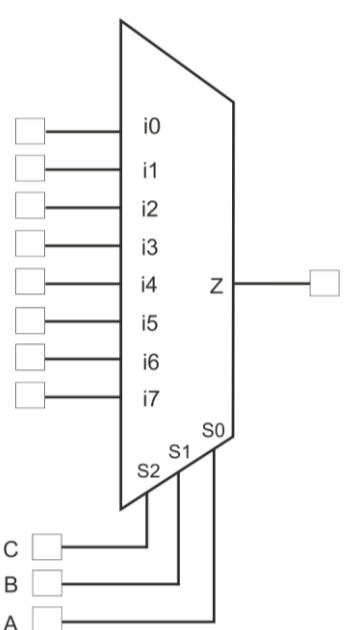
Número: 7
Concepto: Multiplexor como circuito de votación.
Descripción: En este ejercicio, con la utilización de un multiplexor, se deberá completar según las condiciones del enunciado.
Objetivo: Mostrar que cualquier circuito de 3 entradas cabe en una 3-LUT.
Figura: 

Tabla Ejercicios de la aplicación 7: Ejercicio 7

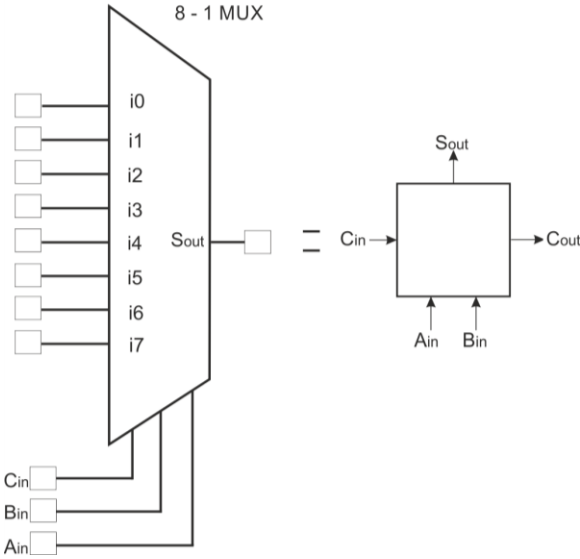
Número: 8
Concepto: Multiplexor para realizar operaciones aritméticas.
Descripción: En este ejercicio, se deberá completar el bit de salida de un circuito full-adder.
Objetivo: Mostrar que cualquier circuito de 3 entradas cabe en una 3-LUT.
Figura: 

Tabla Ejercicios de la aplicación 8: Ejercicio 8

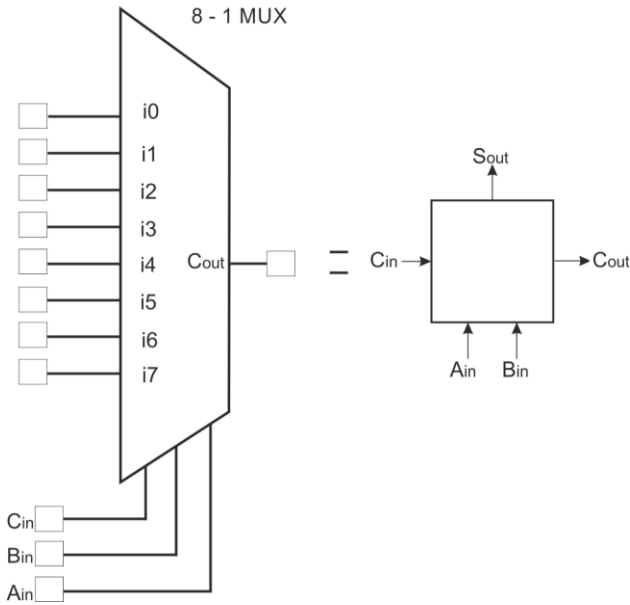
Número: 9
Concepto: Multiplexor como Look-Up Table de 3 entradas.
Descripción: En este ejercicio, se deberá completar la salida de un circuito full-adder, pero teniendo sólo en cuenta el bit carry-out del sumador.
Objetivo: Mostrar que cualquier circuito de 3 entradas cabe en una 3-LUT.
Figura: 

Tabla Ejercicios de la aplicación 9: Ejercicio 9

Número: 10
Concepto: Multiplexor como decodificador.
Descripción: En este ejercicio, se deberá completar la salida para decodificar determinados números de 3 bits.
Objetivo: Mostrar que cualquier circuito de 3 entradas cabe en una 3-LUT.
Figura: <div data-bbox="646 398 997 1075"> </div>

Tabla Ejercicios de la aplicación 10: Ejercicio 10

Número: 11
Concepto: Memoria ROM como Look-Up Table de 4 entradas.
Descripción: En este ejercicio se pretende mapear un decodificador 7-segmentos a través de una Look-Up Table de 4 entradas.
Objetivo: Concepto de LUT basada en una ROM.
Figura: <div data-bbox="494 1478 1085 1926"> </div>

Tabla Ejercicios de la aplicación 11: Ejercicio 11

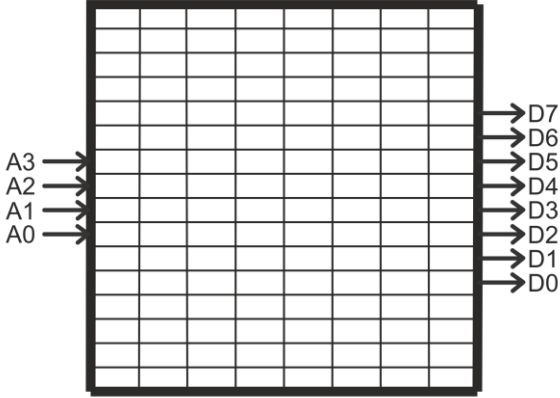
Números: 12, 13 y 14
Concepto: ROM para generar funciones matemáticas complejas.
Descripción: En estos ejercicios, mediante la LUT se realizarán diferentes operaciones según cada enunciado.
Objetivo: Se pretende que el alumno consolide los conceptos sobre Look-Up Table, sabiendo comprender y solucionar distintos problemas mediante su uso.
<p>Figura:</p>  <p>The diagram shows a 4-bit Look-Up Table (LUT) represented as a 16x8 grid. The grid has 16 rows and 8 columns. On the left side, there are four address inputs labeled A3, A2, A1, and A0, each with an arrow pointing to a specific row. On the right side, there are eight data outputs labeled D7, D6, D5, D4, D3, D2, D1, and D0, each with an arrow pointing from a specific column.</p>

Tabla Ejercicios de la aplicación 12: Ejercicios 12, 13 y 14

4 Desarrollo

Este proyecto se ha llevado a cabo siguiendo los siguientes pasos:

- Primeros pasos.
- Herramientas para llevarlo a cabo.
- Partes de una aplicación Android.

4.1 *Primeros pasos*

Antes de empezar a realizar el proyecto, primero se realizaron una serie de acciones:

- Estudio y búsqueda de material sobre JAVA.
- Realización de vídeo tutoriales.
- Realización de pequeñas aplicaciones para familiarizarme con el IDE.

4.2 *Herramientas*

Las herramientas usadas para llevar a cabo este proyecto han sido las siguientes:

- Ordenador personal.
- Smartphone con sistema operativo Android.
- Entorno de desarrollo Android Studio.
- Android SDK Tools.
- Aplicación de diseño gráfico.

4.3 *Partes de una aplicación*

4.3.1 *Actividad*

Una aplicación en Android es básicamente un conjunto de actividades. Cada actividad se implementa en una clase java derivada de la clase base “Activity” de Android.

Las actividades son componentes de la pantalla que interactúan con los usuarios a través de su interfaz de usuario. La interfaz puede hacerse mediante java o un archivo xml.

En java, la interfaz de usuario se crea cuando se ejecuta el método “onCreate” de la actividad.

Las actividades atraviesan una serie de estados desde que se crean hasta su destrucción. Estos estados forman el ciclo de vida de una actividad.

Los distintos estados en los que puede estar una actividad son:

- Activa:
Cuando la actividad está en primer plano.
- En pausa:
Cuando la actividad está parcialmente en segundo plano. La información de estado se mantiene, pero el sistema puede decidir para la ejecución en situaciones extremas.
- Parada:
Cuando otra actividad la cubre por completo.

Los cambios entre estos estados van acompañados de las siguientes llamadas a métodos:

- onCreate()
Cuando la actividad se crea. Se utiliza para crear la interfaz de usuario.
- onStart()
Cuando la actividad se vuelve visible para el usuario.
- onResume()
Cuando la actividad interactúa con el usuario.
- onPause()
Cuando la actividad va a pasar a segundo plano.
- onStop()
Cuando la actividad ya no es visible para el usuario.
- onDestroy()
Cuando la actividad finalmente se destruye.

Veamos un cuadro resumen sobre el ciclo de vida de una actividad:

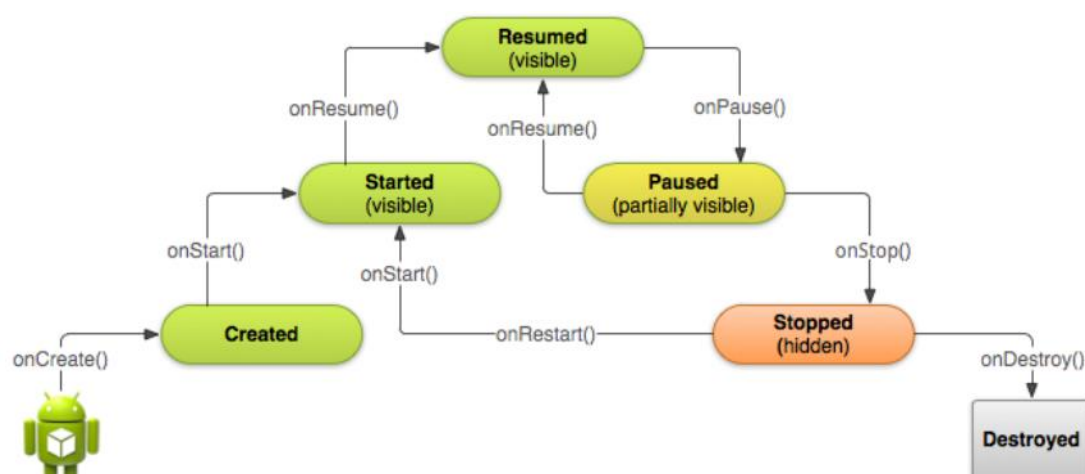


Figura Desarrollo – Partes de una actividad - 1: Ciclo de vida de una actividad Android ¹⁰

4.3.2 Vistas

La interfaz de usuario se implementa con objetos de las clases View (Vistas) y ViewGroup (Contenedores). Son los elementos que aparecen en la pantalla.

En la interfaz de la aplicación hemos utilizado vistas de tipo button y togglebutton (los botones). Y contenedores de tipo LinearLayout que organiza los botones en una sola fila o columna.

4.3.3 La carpeta Manifests

Esta carpeta contiene el fichero AndroidManifest.xml en la raíz del proyecto. Algunos ejemplos de la información que contiene son:

- Las actividades que forman la aplicación:
En nuestro caso, MainActivity.
- El nombre del paquete Java especificado en el atributo “package”.
- La etiqueta de cada actividad, mediante android:label.
- Versiones mínima y máxima de API.

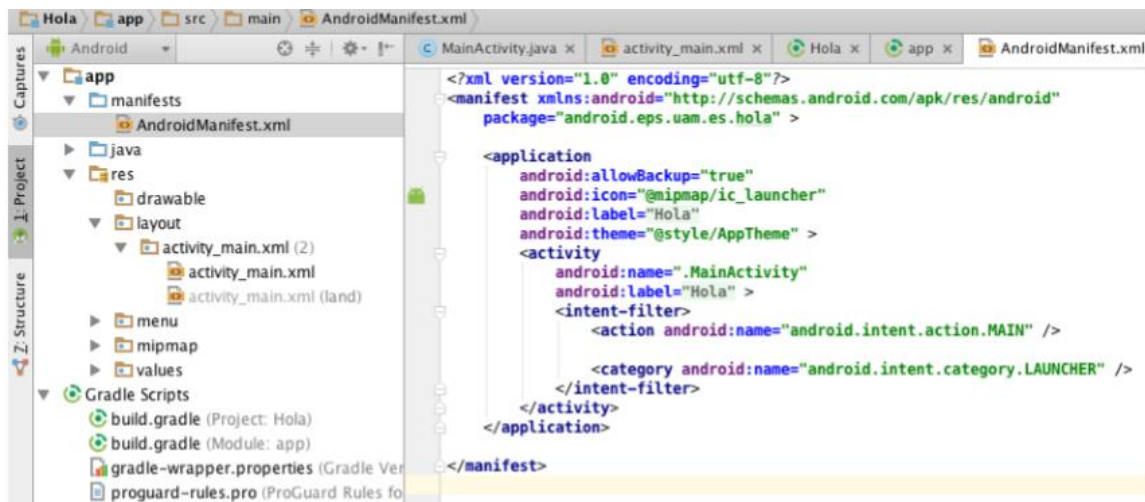


Figura Desarrollo – Carpeta Manifests - 2: Ejemplo de AndroidManifest.xml

4.3.4 La carpeta Java

Contiene el código Java de nuestro proyecto organizado por paquetes

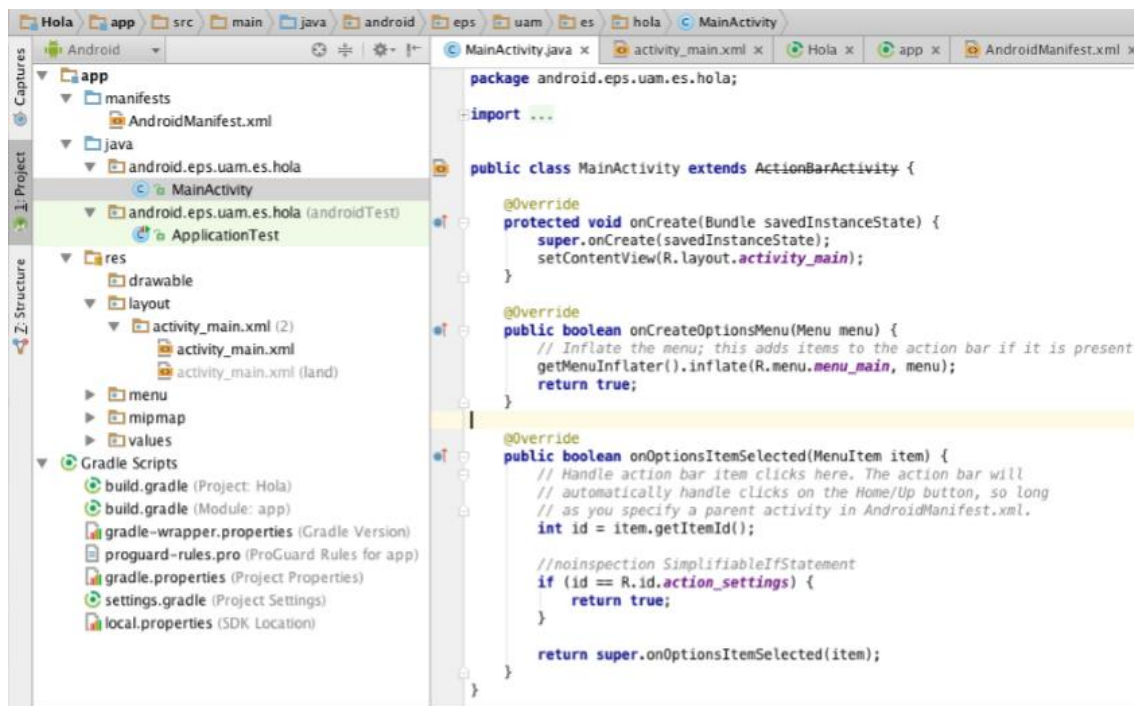


Figura Desarrollo – Carpeta Java- 3: Ejemplo de MainActivity.java

4.3.5 La carpeta Recursos

Contiene los recursos (resources) de la aplicación.

Por ejemplo, las cadenas de caracteres se sitúan en ficheros XML. El código de Java accede a estas cadenas mediante su identificador. Android genera los identificadores de forma automática y se guardan en la clase R.

Se deberían situar cada recurso en las carpetas destinadas para ello, en nuestra aplicación:

- Drawable/
Para ficheros XML o bitmap.
- Layout/
Para ficheros XML que especifican interfaces de usuario.
- Menu/
Para ficheros XML que especifican la interfaz de los menús.
- Values/
 - colors.xml : recursos que especifican colores.
 - dims.xml : recursos que especifican dimensiones.
 - strings.xml : para cadena de caracteres.
 - styles.xml : recursos que especifican estilos.

4.3.6 La carpeta Gradle.scripts

Es la herramienta que utiliza Android Studio para la realización de aplicaciones.

El objetivo es construir un fichero empaquetado .apk a través de los distintos recursos, ficheros java y las bibliotecas.

4.4 Menús de la aplicación

La aplicación está dividida a través de sencillos menús para poder ir a las opciones disponibles. Los menús son los siguientes:

- **Menú principal:**
El menú principal es al que accedemos al iniciar la aplicación. El cual nos muestra las opciones de poder ir al tutorial, menú de ejercicios, envío de ejercicios, o ayuda.
- **Menú ejercicios:**
Se accede desde el menú principal y nos muestra un menú con los diferentes ejercicios disponibles.
- **Menú ayuda:**
Se accede desde el menú principal y nos lleva a otro menú en el que tenemos diferentes opciones como salvar, cargar o envío de ejercicios, o completar una tabla.
- **Menú botón Android:**
Está disponible en todo momento en la aplicación para llevarnos a las opciones de ayuda o acerca de (about), que recoge información de la aplicación.

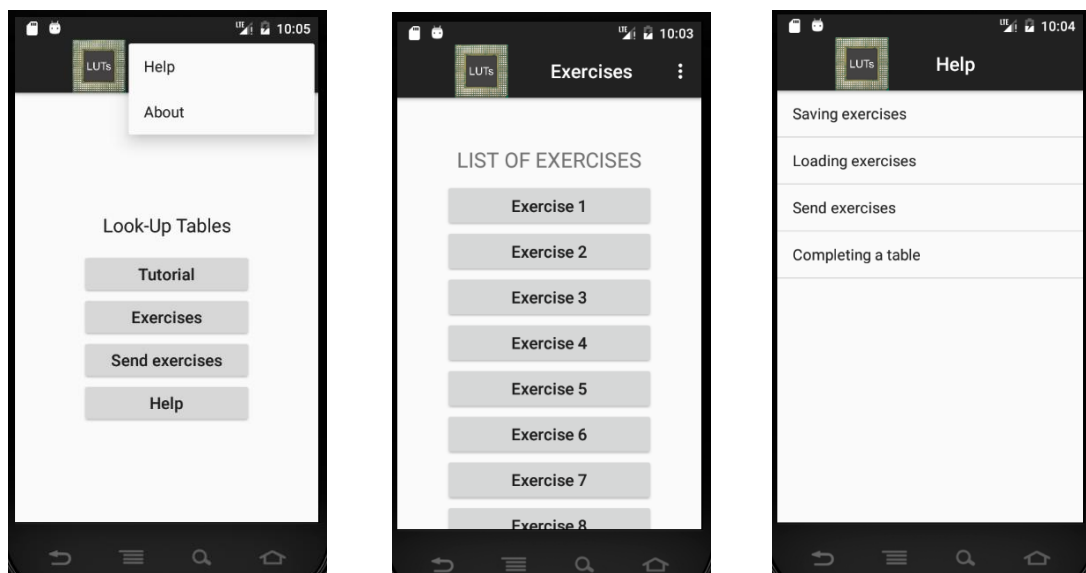


Figura Desarrollo – Menús de la aplicación - 4: Menús de la aplicación

4.4.1 Tutorial

La aplicación cuenta con un tutorial en forma de diapositivas o imágenes sobre el tema de las Look-Up Tables de las asignaturas “Circuitos Electrónicos Digitales” y “Dispositivos Integrados Especializados”.

Se accede desde el menú principal como se ha mencionado anteriormente, y bastará con desplazar el dedo de izquierda a derecha según se quiera avanzar o retroceder de diapositiva.

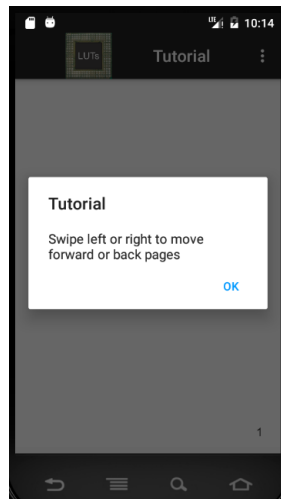


Figura Desarrollo – Tutorial - 5: Tutorial de la aplicación

4.4.2 Acerca de

Mediante el botón menú de Android disponible en todo momento en la aplicación, podemos acceder a dos opciones, ayuda, mencionada anteriormente, y about o acerca de.

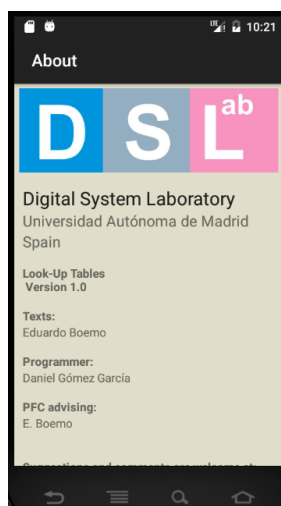


Figura Desarrollo – Acerca de - 6: About de la aplicación

Esta opción nos muestra información en particular de la aplicación, como la versión de la misma, el laboratorio y la Universidad donde se hizo. Se ha seguido el formato de las otras aplicaciones desarrolladas en el DSLab.

4.5 Diseño y desarrollo de los ejercicios

Para llevar a cabo el desarrollo de estos ejercicios es necesario realizar la vista de la actividad y la propia actividad del ejercicio.

4.5.1 Vista de la actividad

Desarrollamos los ejercicios a través de un archivo .xml

En primer lugar, hemos definido los contenedores pues determinan la estructura visual de las vistas.

El primero de ellos es el ScrollView, ya que nos permite mostrar el contenido que no cabe en una sola pantalla mediante una barra vertical de scroll. Este contenedor sólo puede tener un elemento, que hemos seleccionado un contenedor LinearLayout que contenga el resto de vistas de la pantalla.

Después hemos elegido el contenedor TableLayout para organizar los elementos en filas y columnas. Cada fila se define mediante un elemento TableRow que puede tener 0 o más elementos, y el número de columnas es el número de elementos de una fila.

Dependiendo del ejercicio hemos determinado un mayor o menos número de botones de dos tipos en un mayor o menor número de filas y columnas:

- **ToggleButton:** tiene dos estados, 0 y 1, por defecto empieza en 0, y se puede cambiar de un estado a otro indistintamente al pulsarlo.
- **Button:** Definidos siempre al final de cada ejercicio para activar las distintas funcionalidades de comprobación, solución, reseteo o salvado del ejercicio.

En los ejercicios 12, 13 y 14, hemos seleccionado elementos de texto en algunas filas para que tuviese la apariencia deseada.

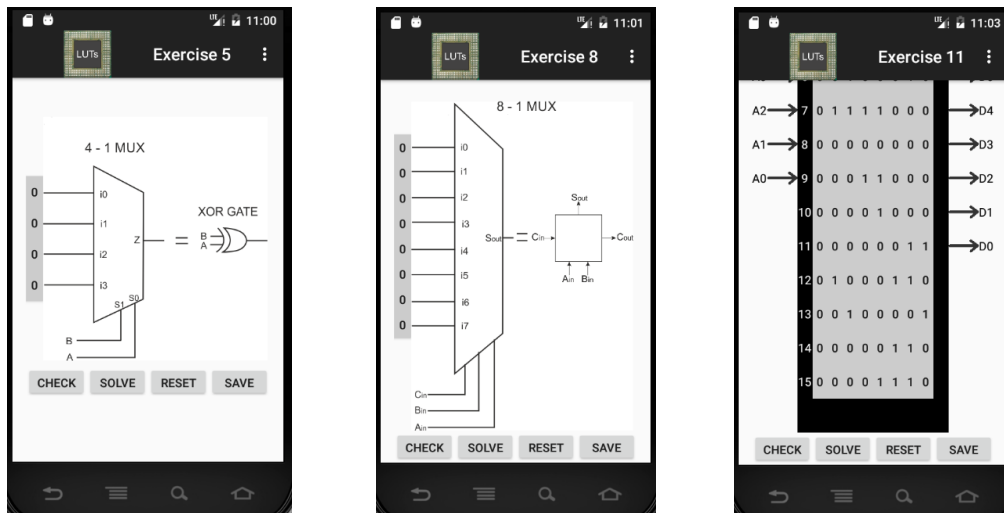


Figura Vista de la actividad - 7: Apariencia de los ejercicios de la aplicación

4.5.2 Actividad

En cada ejercicio tenemos que ir registrando cada dato introducido por el usuario. Para ello tenemos que crear primero los botones de tipo `ToggleButton` con su propio identificador para conocer cual pulsa el usuario en cada momento.

Después crearemos dos tipos de vectores, un vector de tipo `ToggleButton` que contiene todos los botones `ToggleButton` definidos y que lo que modifiquemos aquí, se modificará en la vista.

Y otro vector de tipo `Boolean` para copiar los datos del primer vector mediante el método `Guardar_TB` y la función `isChecked` que me da el estado del `ToggleButton`, y así poder manejarlos sin que cambie la vista.

Una vez tenemos los datos introducidos por el usuario registrados procedemos a las funciones de comprobación, solución, reseteo o guardado del ejercicio.

4.6 Gestión de datos

4.6.1 Método de preferencias

Las preferencias son archivos XML en los que las actividades guardan la información de la aplicación. Cada preferencia tiene un identificador y un valor único para ser localizado.

Cada actividad tiene que declarar un objeto de tipo “SharedPreferences” y a continuación decir el nombre del archivo donde se guardarán los datos. Para escribir los datos en dicho archivo, tendremos que declarar un editor.

Para cargar los datos del archivo, usaremos métodos como `getIn` o `getString`.

En la aplicación se usa el método de preferencias en:

- Los enunciados de los problemas:
Sólo se guarda el número de ejercicio para cargar o eliminar dicho ejercicio.
- Los diferentes problemas:
Cuyas soluciones del usuario son guardadas de manera que al volver al ejercicio sin hacer un reset antes, permanezcan los datos.

4.6.2 Check

Se realizan comprobaciones de los datos introducidos por el usuario con una solución única disponible.

Las comparaciones se realizan por filas comparando el vector introducido por el usuario con el vector solución, además en caso de haber errores muestra un mensaje de texto formado por una cadena de caracteres en forma de ayuda indicando dónde hay fallos en el ejercicio.



Figura Gestión de datos - Check - 8: Mensaje de comprobación de ejercicio

4.6.3 Solve

Se muestra la solución correcta del ejercicio. El procedimiento es el mismo que la opción de resetear, solo que en vez de dar valores de null, mostramos la solución.

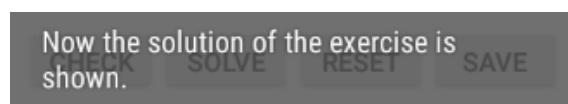


Figura Gestión de datos - Solve - 9: Mensaje de mostrar solución en ejercicio

4.6.4 Reset

Antes de resetear se pregunta al usuario si está seguro de querer borrar los datos introducidos, si es así, mediante el método `setChecked` pondremos todos los botones `ToggleButton` a 0.

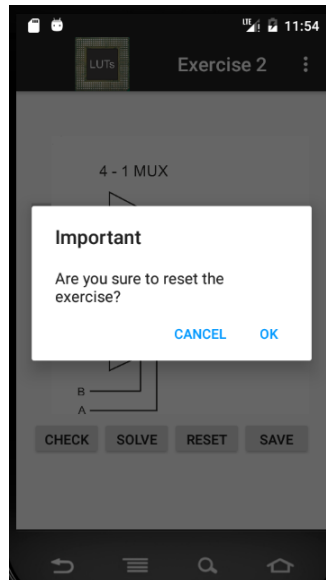


Figura Gestión de datos - Reset - 10: Mensaje de reseteo de ejercicio

4.6.5 Save

Se definen todos los vectores necesarios para guardar los datos de cada ejercicio. La vista de la actividad muestra un diálogo en el cual pregunta al usuario el nombre con el que guardar el ejercicio.

El diálogo tiene una cabecera en la cual indica el número de ejercicio a guardar seguido de EM si el archivo se guardará en memoria externa o IM si el archivo será guardado en la memoria interna del dispositivo.

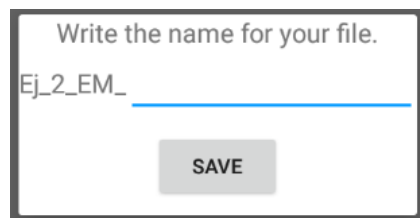


Figura Gestión de datos - Save - 11: Mensaje de guardado de ejercicio

4.6.6 Load or delete

El acceso a esta actividad es desde cada enunciado de los diferentes ejercicios que forman la lista.

Lo primero que es necesario es tener guardado un ejercicio previamente, sino nos mostrará un mensaje diciendo que no hay archivos salvados.

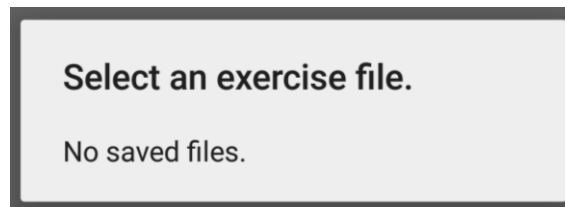


Figura Gestión de datos – Load or delete - 12 : Mensaje de cargar un ejercicio

En caso de tener varios archivos guardados, tendremos que seleccionar el que queramos ya sea para eliminar o cargar un ejercicio, con la diferencia de que eliminar se pueden eliminar varios archivos a la vez, pero cargar sólo se puede cargar un único archivo, apareciendo un mensaje de error si queremos cargar más de un archivo a la vez.

Si elegimos la opción de eliminar archivo, nos mostrará una nueva alerta preguntándonos si estamos seguros de que lo queremos eliminar, ya que una vez digamos que sí, no se podrá recuperar.

Al cargar los datos de un archivo, se sobrescriben a los que hubiese anteriormente, ya que esos datos estaban guardados en el archivo de preferencias y ahora se guardarán ahí.

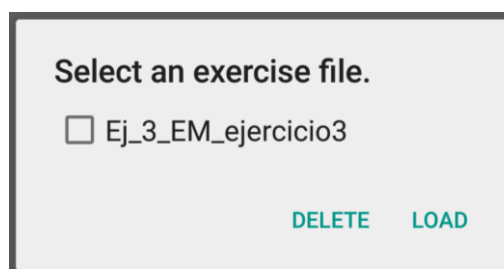


Figura Gestión de datos – Load or delete - 13: Mensaje de cargar o eliminar un ejercicio

4.6.7 Send exercises

A esta opción se accede desde el menú principal de inicio.

Nos facilita la opción del envío único o múltiple de archivos, por diversos medios a elección del usuario.

Una vez se ha seleccionado el método de envío, se adjuntan los archivos, se rellena de manera automática tanto el asunto como el cuerpo del mensaje con unas instrucciones para que el receptor conozca dónde serán descargados los archivos en su terminal, y qué pasos tienen que hacer para poder cargarlos de manera satisfactoria.

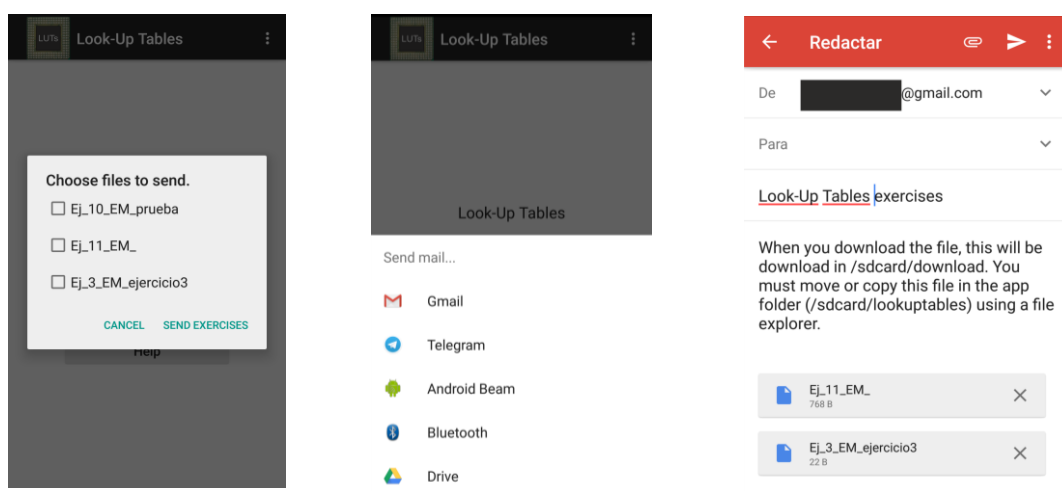


Figura Gestión de datos – Send exercises - 14: Proceso de envío de un ejercicio.

4.7 Adaptación de pantallas

Para que la aplicación pueda ser visualizada de forma correcta en los distintos tipos de dispositivos con sus diferentes tamaños y densidades de pantallas tenemos que adaptar las vistas.

Para crear interfaces de usuario en Android tenemos que especificar distintos ficheros de diseño para cada tamaño de pantalla y distintas versiones de imagen para cada densidad. De cara a que no haya problemas en ver las imágenes borrosas si se amplían o contraen los píxeles.

Hemos creado en XML distintas carpetas, en concreto, para los tamaños normal (de 3 a 4.5 pulgadas) y large (de 4.5 a 7 pulgadas), ya que, como explicamos anteriormente son los tamaños más usados en dispositivos Android. Lo que cambia en los archivos para la correcta visualización son los tamaños de fuente y los márgenes según el tamaño de pantalla.

Y lo mismo sucede con las densidades, usándose versiones diferentes de cada imagen dependiendo de la densidad requerida.

5 Integración, pruebas y resultados

5.1 Google Play Store

Una vez que hemos comprobado que la aplicación funciona correctamente y no tenemos errores en ella, procederemos a su publicación. Ya que ese era el objetivo del proyecto, la publicación de la aplicación para que los alumnos puedan hacer uso de ella.

5.2 Preparación

Lo primero que se debe hacer es empaquetar la aplicación, es decir, generar el archivo APK (Application Package File) para su publicación.

El gestor de paquetes (package manager) de un dispositivo Android no instalará una aplicación que no haya sido firmada.

Durante el desarrollo se utiliza una llave de depuración, pero para publicar la aplicación es necesaria una llave privada que nos identifique, y sólo nosotros podamos modificar la aplicación.

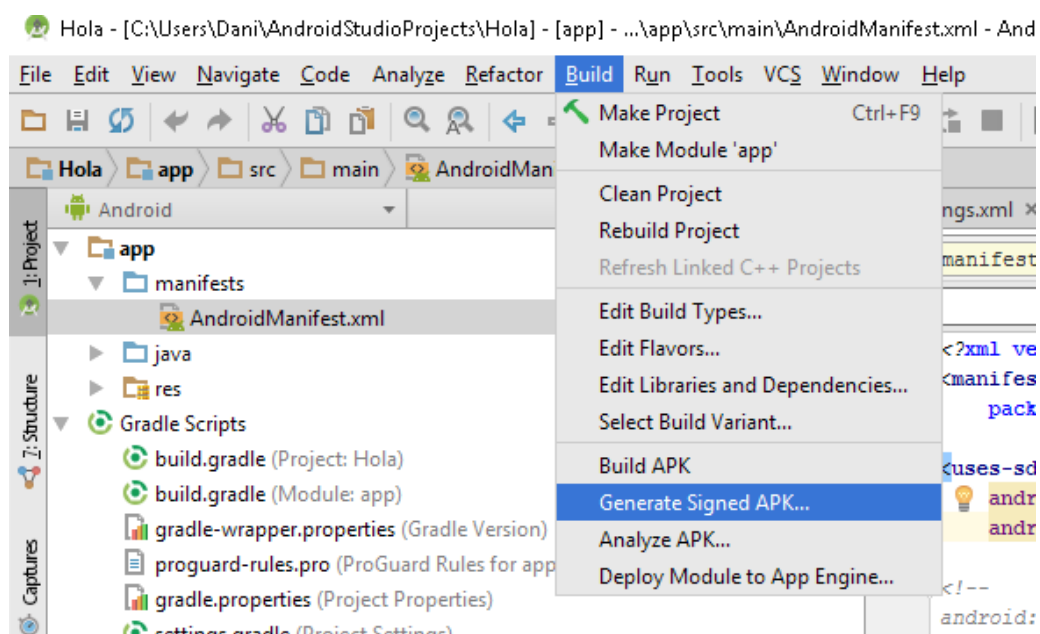
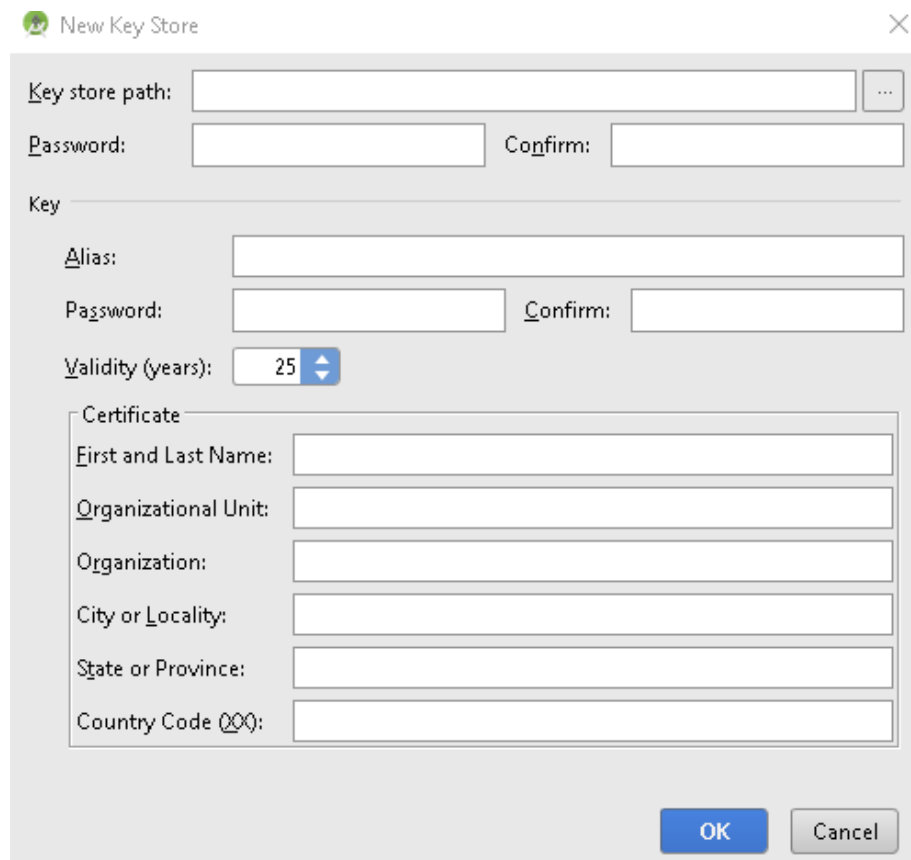


Figura Integración, pruebas y resultados– Preparación - 1: Proceso de generación APK

Deberemos crear un nuevo almacén de claves (Key Store) así como nueva clave (Key). Hay que rellenar los siguientes campos:



The image shows a 'New Key Store' dialog box. It has a title bar with a close button. The main area contains the following fields and controls:

- Key store path:** A text input field followed by a browse button (...).
- Password:** A text input field.
- Confirm:** A text input field.
- Key:** A section header followed by:
 - Alias:** A text input field.
 - Password:** A text input field.
 - Confirm:** A text input field.
 - Validity (years):** A spinner control set to 25.
- Certificate:** A section header followed by a group box containing:
 - First and Last Name:** A text input field.
 - Organizational Unit:** A text input field.
 - Organization:** A text input field.
 - City or Locality:** A text input field.
 - State or Province:** A text input field.
 - Country Code (XXX):** A text input field.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom right.

Figura 5-2-2: Elección de claves de generación APK

En APK destination folder indicaremos el lugar donde queremos guardar nuestro archivo APK. El nombre por defecto con el que será guardado es app-release.apk que se puede modificar y poner el nombre que queramos.

Una vez tenemos el archivo apk, se puede enviar por correo para probarlo en distintos móviles, ya que funciona como un ejecutable .exe

Hay que habilitar la opción de permitir instalación de aplicaciones de orígenes desconocidos en el apartado de seguridad.

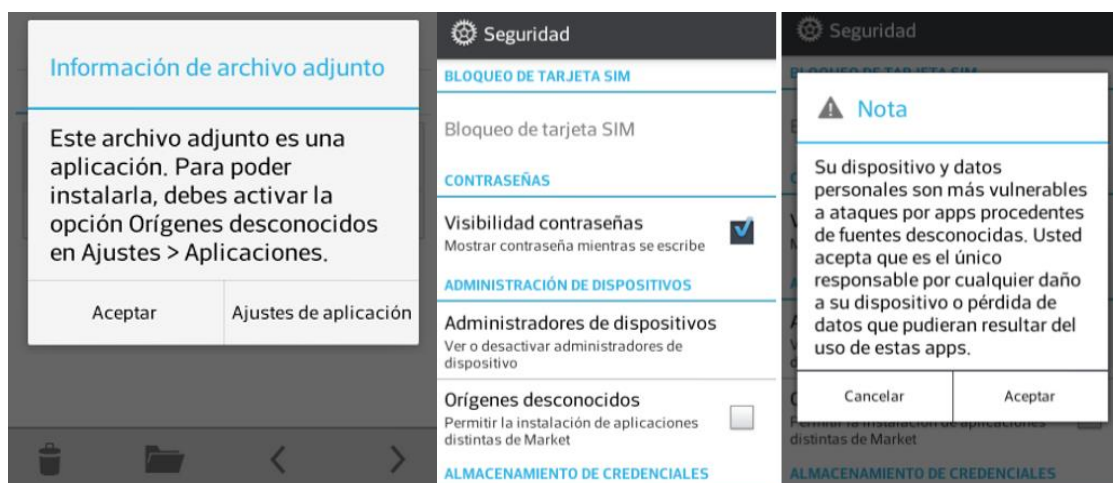


Figura Integración, pruebas y resultados– Preparación - 2: Proceso de instalación archivo APK

5.3 Cuenta de desarrollador

Es necesaria una cuenta de desarrollador para poder publicar una aplicación en Google Play Store. El precio a pagar por una cuenta si no se dispone de ella son US25\$ y no caduca, al contrario que en App Store que cuesta US99\$ por año.

En nuestra aplicación utilizaremos la cuenta del laboratorio DSLab donde se ha desarrollado este proyecto, que tiene por nombre de desarrollador: “DSLab UAM”.

5.4 Publicación

Tenemos que acceder a la web “Google Play Android Developer Console” con la cuenta de desarrollador.

Una vez dentro hay que pulsar la opción de añadir nueva aplicación, adjuntar el archivo apk y dar una serie de datos relativos de la aplicación:

- Título.
- Idioma.
- Descripción.
- Icono de la aplicación.
- Distintas capturas de pantalla.
- Ámbito docente.
- Política de privacidad.
- Precio y distribución, en nuestro caso gratuita y al mayor número de países(142).

6 Conclusiones y trabajo futuro

6.1 Conclusiones

El objetivo inicial de este proyecto era el desarrollo de una aplicación Android que pudiera facilitar a los alumnos de la Escuela su aprendizaje y desarrollo de conocimientos sobre Look-Up Tables de las asignaturas “Circuitos Electrónicos Digitales” y “Dispositivos Integrados Especializados”.

Y una vez terminado el proyecto podemos decir que se han cumplido los objetivos, pues principalmente era el desarrollo de la aplicación cumpliendo una serie de requisitos expuestos más adelante. Los resultados en los alumnos se verán transcurridos con el paso del tiempo.

Para la realización del proyecto se han tenido que cumplir una serie de requisitos:

- Tutorial respecto a las Look-Up Tables.
- Una colección de 14 ejercicios sobre Look-Up Tables.
- Funciones de comprobación, reseteo, solucionado y guardado de los ejercicios.
- Funciones de cargar o eliminar ejercicios guardados.
- La posibilidad de enviar los ejercicios mediante correo electrónico.
- Manuales de ayuda al usuario.
- Adaptación de las vistas a diferentes tamaños y densidades de pantalla.

Durante el transcurso del proyecto se han adquirido conocimientos sobre los siguientes temas:

- Programación en Android basado en Java.
- Programación XML.
- Utilización de herramientas de desarrollo gráfico.
- Utilización del entorno de desarrollo Android Studio.
- Publicación de una aplicación.

6.2 Trabajo futuro

Con la expansión de las nuevas tecnologías y su uso enfocado al ámbito docente, se podría aprovechar esta aplicación de cara a ayudar al mayor número de alumnos para favorecer su ampliación y consolidación de conocimientos.

Algunos ejemplos de mejoras serían:

- Idiomas: Es un punto importante ya que con un mayor número de idiomas podría llegar a un mayor número de usuarios, dejando de ser una barrera para el aprendizaje.
- Ampliar la colección de ejercicios con nuevos conceptos y variantes de los ya existentes.
- Mejorar la interfaz de usuario.
- Incluir un sistema de comunicación entre usuarios de cara a resolver dudas de una manera más rápida y colaborativa.
- Incluir una serie de test con múltiple respuesta de cara a consolidar conocimientos en los alumnos.
- Incluir un sistema de puntuación con seguimiento para mejorar la progresión de los mismos.
- Analizar el consumo de la batería, enfocado a una mejor optimización.

Referencias

- [1] Horton, John. “Android Programming for Beginners Learn all the Java and Android skills you need to start making powerful mobile application”. Editor Packt Publishing. 2015
- [2] Sznajdleder, Pablo Augusto.”Java a fondo curso de programación”. Editor Marcombo. 2016
- [3] McCracken, Scott.”Android curso de desarrollo de aplicaciones “. Editor Inforbook's. 2012
- [4] Ribas Lequerica, Joan.” Desarrollo de aplicaciones para Android “. Editor Anaya Multimedia. 2016
- [5] Tomás Gironés, Jesús.”El Gran Libro de Android”. Editos Marcombo. 2016
- [6] <https://play.google.com/store>
- [7] <https://developer.android.com/index.html>
- [8] <http://www.androidauthority.com/android-version-distribution-748439/>
- [9] <http://www.aprendeandroid.com/>
- [10] <https://elbauldelprogramador.com/curso-programacion-android/>
- [11] <http://www.cursodejava.com.mx/>
- [12] <https://androidstudiofaqs.com/tutoriales/programar-en-android-studio>
- [13] <http://www.sgoliver.net/blog/curso-de-programacion-android/indice-de-contenidos/>
- [14] <http://www.tutorialesprogramacionya.com/javaya/androidya/androidstudioya/>

Glosario

API	Application Programming Interface
APK	Application Package File
BitMap	Mapa de bits
CED	Circuitos Electrónicos Digitales
DSLAb	Digital System Lab
EPS	Escuela Politécnica Superior
IDE	Integrated Development Environment
LUT	Look-Up Table
Kernel	Núcleo del Sistema Operativo
ROM	Memory Only Read
SO	Sistema Operativo
UAM	Universidad Autónoma de Madrid
UI	Interfaz de Usuario

Anexos

A Clases y métodos Java

Clase	Descripción
Tutorial	Se utiliza para ver correctamente el avance y retroceso de las diapositivas.
ImageAdapter	Se utiliza para cargar las imágenes usadas en el tutorial.
Ejercicios	Se utiliza para ir a un ejercicio en concreto de la lista de problemas.
Ayuda1	Se utiliza para ver las diferentes ayudas disponibles.
Ayuda2	Se utiliza para ver la ayuda seleccionada.
About	Se utiliza para mostrar información acerca de la aplicación.
EnunciadoX	Se utiliza para el comienzo de un ejercicio, también permite la carga o borrado de una solución almacenada anteriormente.
EjercicioX	Se utiliza para la resolución del ejercicio seleccionado.
CargarSolucionEnArchivo	Se utiliza para cargar soluciones guardadas anteriormente y mostrarlas.
GuardarSolucionEnArchivo	Se utiliza para guardar los datos introducidos por el usuario en un archivo.
MainActivity	Se utiliza para representar el menú principal incluyendo la opción del envío de soluciones por correo.

Tabla Anexo 1: Clases Java de la Aplicación

Método	Descripción
onCreate()	Se utiliza para desarrollar todas las opciones al inicio de una actividad.
onClick()	Se utiliza cuando el usuario pulsa algún objeto de la pantalla.
guardar_TB()	Se utiliza para guardar los datos introducidos en los ToggleButton.
save()	Se utiliza para guardar la solución introducida por el usuario.
comprobación()	Se utiliza para comprobar si el resultado introducido por el usuario es correcto.
ayuda()	Se utiliza para cargar la solución en los ejercicios de tipo tabla.
resetear()	Se utiliza para el borrado de los datos introducidos por el usuario.
onKeyDown()	Se utiliza para ir a la actividad anterior a la que está, mediante back.
lanzarejX()	Se utiliza para iniciar el ejercicio especificado.
lanzarTutorial()	Se utiliza para iniciar el tutorial.
lanzarEnviarEjercicios()	Se utiliza para iniciar el envío de soluciones guardadas por el usuario.
onCreateOptionsMenu()	Se utiliza para crear un menú de opciones en el botón de Android.
guardarPreferencias()	Se utiliza para guardar los datos introducidos por el usuario en el archivo de preferencias.
cargarPreferencias()	Se utiliza para cargar el archivo de preferencias que contienen los datos de su último uso.

Tabla Anexo 2: Métodos Java de la Aplicación

B Imágenes de los ejercicios de la aplicación

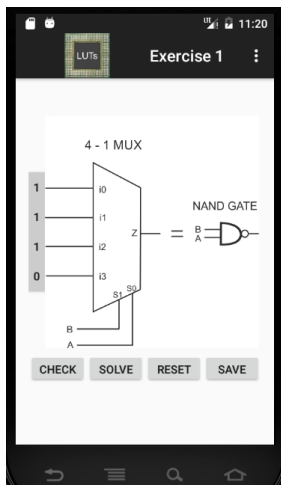


Figura 0-1: Ejercicio 1

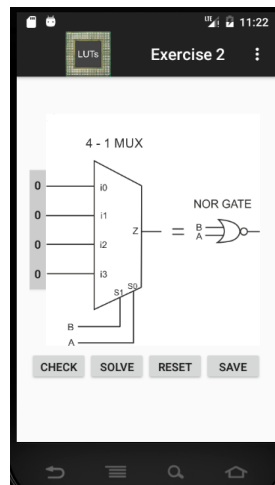


Figura 0-2: Ejercicio 2

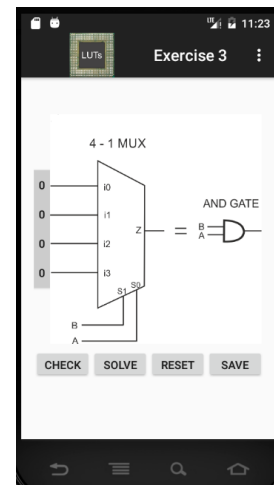


Figura 0-3: Ejercicio 3

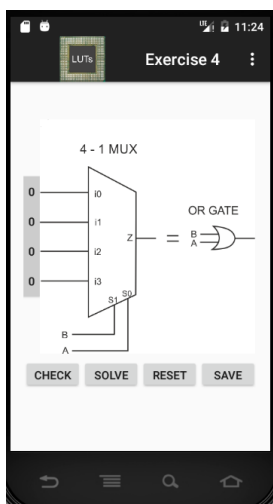


Figura 0-4: Ejercicio 4

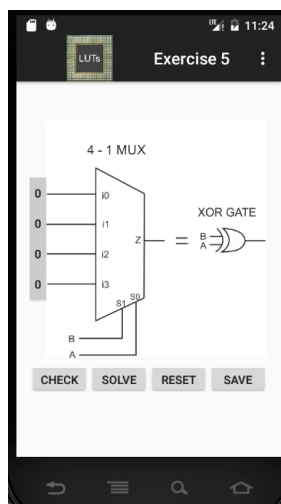


Figura 0-5: Ejercicio 5

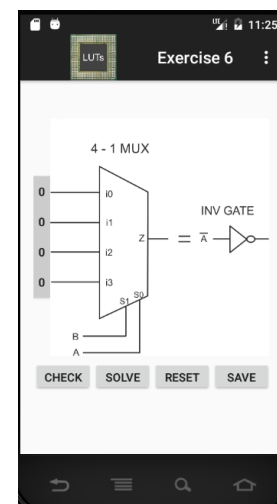


Figura 0-6: Ejercicio 6

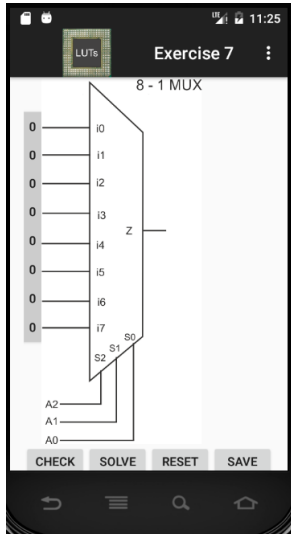


Figura 0-7: Ejercicios 7 y 10

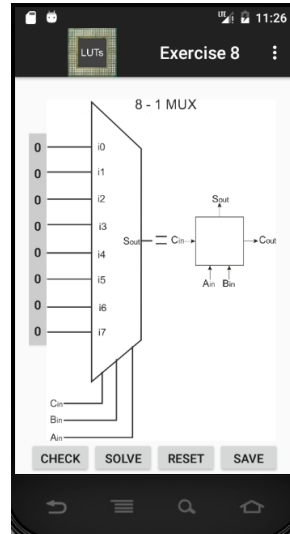


Figura 0-8: Ejercicio 8

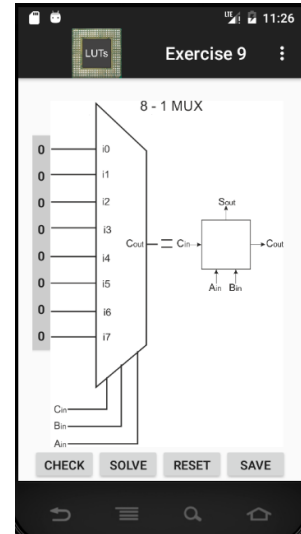


Figura 0-9: Ejercicio 9

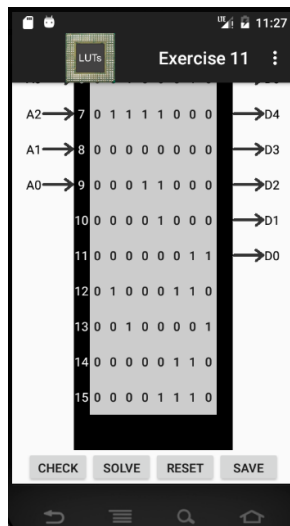


Figura 0-11: Ejercicio 11,12,13 y 14

C Fuentes bibliográficas

¹ IDC, Nov 2016 : <http://www.idc.com/promo/smartphone-market-share/os>

² Statista, 2017: <https://www.statista.com/statistics/266219/global-smartphone-sales-since-1st-quarter-2009-by-operating-system/>

³ <http://www.androidauthority.com/android-version-distribution-748439/>

⁴ <http://www.androidauthority.com/android-version-distribution-748439/>

⁵ <https://developer.android.com/about/dashboards/index.html>
https://developer.android.com/guide/practices/screens_support.html?hl=es-419

⁶ <https://developer.android.com/about/dashboards/index.html>
https://developer.android.com/guide/practices/screens_support.html?hl=es-419

⁷ https://developer.android.com/guide/practices/screens_support.html

⁸ https://developer.android.com/guide/practices/screens_support.html

⁹ Internet World Stats, Marzo 2017: <http://www.internetworldstats.com/stats7.htm>

¹⁰ <https://developer.android.com/training/basics/activity-lifecycle/starting.html>